

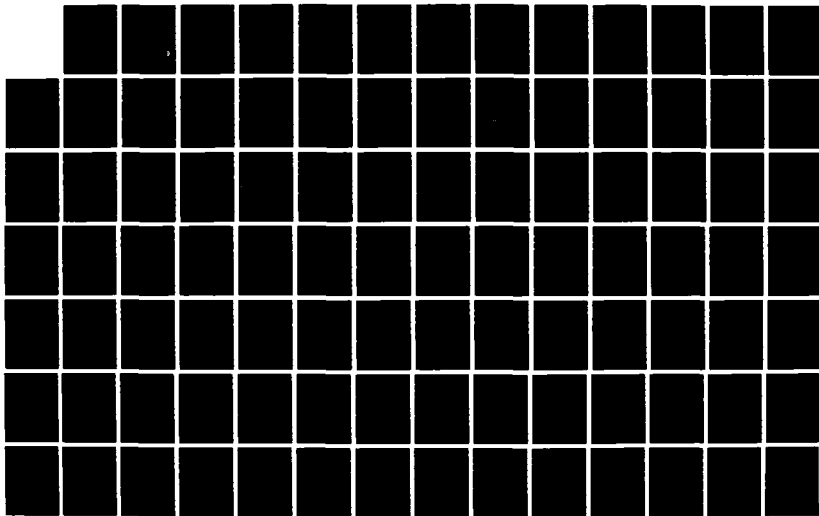
AD-A190 761

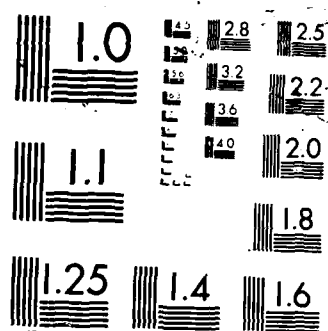
MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED
TO FLEXIBLE SPACES. (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R W LASHLEE
DEC 87 AFIT/GE/ENG/87D-36 F/G 22/4

1/3

UNCLASSIFIED

NL





AD-A190 761



MOVING-BANK MULTIPLE MODEL ADAPTIVE
ESTIMATION APPLIED TO FLEXIBLE
SPACESTRUCTURE CONTROL

THESIS

Robert W. Lashlee, Jr.
Captain, USAF

AFIT/GE/ENG/87D-36

DTIC
ELECTE
MAR 31 1988
S E D

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved
for public release and sale in
distribution is unlimited.

88 3 30 069

1

MOVING-BANK MULTIPLE MODEL ADAPTIVE
ESTIMATION APPLIED TO FLEXIBLE
SPACESTRUCTURE CONTROL

THESIS

Robert W. Lashlee, Jr.
Captain, USAF

AFIT/GE/ENG/87D-36

DTIC
ESTIMATE
S
FI

This document has been approved
for public release and sale; its
distribution is unlimited.

MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION
APPLIED TO FLEXIBLE SPACESTRUCTURE CONTROL

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Robert W. Lashlee, Jr., B.S.
Captain, USAF

December 1987

Accession For	
NTIS	CRA&I
DTIC	TAB
Unannounced	
Justification	
By	
Distribution	
Availability	
Dist	DTIC
A-1	COPY INSPECTED 4

Approved for public release; distribution unlimited

Preface

The purpose of this thesis was to demonstrate the feasibility of the moving-bank multiple model adaptive estimation algorithms as applied to flexible spacestructure control. Moving-bank multiple model adaptive estimation and control is an attempt to reduce the computational loading associated with the implementation of a full-scale multiple model adaptive estimator/controller. The results of this thesis showed that the performance of the moving-bank controller converged to the performance of a benchmark controller (ideal performance). Also, the moving-bank controller substantially outperformed a fixed bank controller with a discretization that covered the range of parameter variation.

I would like to thank Dr. Peter Maybeck for his guidance, wisdom, and undying enthusiasm throughout my thesis effort. I also wish to thank my children, Robert and Tiffanee, for their understanding and patience, even at their young ages. Finally, I would like to thank my wife, Tamee, for all the love, understanding and endless support throughout my AFIT assignment.

--- Bob Lashlee

Table of Contents

	Page
Preface	ii
List of Figures	vi
List of Tables	ix
Abstract	x
I. Introduction	1
1.1. Background	2
1.2. Problem	9
1.3. Scope	9
1.4. Approach	10
1.5. Summary	18
II. Algorithm Development	19
2.1. Introduction	19
2.2. Bayesian Estimation Algorithm Development	19
2.3. Moving Bank Algorithm Development . . .	27
2.3.1. Weighted Average	27
2.3.2. Sliding the Moving Bank . . .	29
2.3.2.1. Residual Monitoring	29
2.3.2.2. Probability Monitoring	30
2.3.2.3. Parameter Position Estimate Monitoring	30
2.3.2.4. Parameter Position and Velocity Estimate Monitoring	31
2.3.3. Bank Contraction and Expansion	31
2.3.4. Initialization of New Elemental Filters	35
2.4. Controller and Estimator Design	37
2.5. Ambiguity Function Analysis	41
2.6. Summary	45

	page
III. Rotating Two-Bay Truss Model	47
3.1. Introduction	47
3.2. Second Order and State Space Form Models	47
3.3. Modal Analysis	51
3.4. Two-Bay Truss	53
3.4.1. Introduction	53
3.4.2. Background	54
3.4.3. Two-Bay Truss Construction	57
3.4.4. Sensors and Actuators	59
3.4.5. Physical System Parameter Uncertainty	60
3.5. State Reduction	61
3.5.1. Introduction	61
3.5.2. Development	62
3.5.3. Order Reduction Selection	67
3.6. Summary	67
IV. Simulation Plan	70
4.1. Introduction	70
4.2. Monte Carlo Analysis	70
4.3. Software Description	75
4.3.1. Introduction	75
4.3.2. Preprocessor	76
4.3.3. Primary Processor	77
4.3.4. Post Processor	78
4.3.5. Ambiguity Function Analysis	78
4.4. Simulation Plan	79
4.4.1. Noise Level Determination	79
4.4.2. Two-State Model	80
4.4.3. Six-State Model	81
4.4.4. Space Discretization	81
4.4.5. Measurement Covariance and Dynamics Noise Sensitivity	83
4.4.6. Controller Design: Need for Cross Terms in the Quadratic Cost Function	84
4.4.7. State Weighting and Control Weighting Matrix Tuning	86
4.4.8. Controller Study	87

	page
4.5. Summary	88
V. Results	89
5.1. Introduction	89
5.2. Measurement Covariance Noise and Dynamics Noise Strength Determination	92
5.3. Two-State Model	93
5.4. Six-State Filter	100
5.5. Space Discretization	115
5.6. Measurement and Dynamic Noise Sensitivity	131
5.7. Controller Design: Need for Cross Terms in the Quadratic Cost	140
5.8. State Weighting and Control Weighting Matrix Tuning	141
5.9. Controller Study	144
5.10. Summary	195
VI. Conclusions and Recommendations	197
6.1. Introduction	197
6.2. Conclusions	197
6.3. Recommendations	199
Appendix A: LQG Controller Development	201
Appendix B: Rotating Two-Bay Truss System Matrices	204
Bibliography	211
vita	216

List of Figures

	page
1-1. Multiple Model Adaptive Estimator . . .	3
1-2. Moving-Bank Multiple Model Adaptive Estimator	6
1-3. Bank Changes a. Move, b. Expansion . .	8
2-1. Bank Discretization: a. Coarse, b. Fine	32
2-2. Probability Weighting of Sides	34
3-1. Two-Bay Truss Model	55
3-2. Rotating Two-Bay Truss Model	56
4-1. System Estimator and Controller Simulation	73
5-1. Two-State Model Bank Location Time History; True Parameter (1,1)	97
5-2. True Parameter (9,2)	98
5-3. True Parameter (2,9)	99
5-4. Six-State Model Bank Location Time History; (Old Discretization Level) True Parameter (2,7)	102
5-5. True Parameter (6,4)	103
5-6. True Parameter (9,9)	104
5-7. True Parameter (3,3)	105
5-8. State Estimation Errors (Old Discretization Level) True Parameter (2,7)	106
5-9. True Parameter (9,9) State Estimation Errors	109
5-10. True Parameter (7,6) State Estimation Errors	112

	page
5-11. Old and New Space Discretization Levels	116
5-12. Six-State Model Bank Location Time History; (New Discretization Level) True Parameter (2,7)	118
5-13. True Parameter (6,4)	119
5-14. True Parameter (9,9)	120
5-15. True Parameter (3,3)	121
5-16. State Estimation Errors (New Discretization Level) True Parameter (2,7)	122
5-17. True Parameter (9,9)	125
5-18. True Parameter (7,6)	128
5-19. Six-State Model Bank Location Time History; R1, R2, R3 = maximum value True Parameter (2,7)	134
5-20. True Parameter (6,4)	135
5-21. True Parameter (9,9)	136
5-22. True Parameter (3,3)	137
5-23. Six-State Model Bank Location Time History; R1, R2, R3 > maximum value True Parameter (1,1)	138
5-24. True Parameter (9,9)	139
5-25. Multiple Model Adaptive Controller . . .	147
5-26. Benchmark Controller; Full Simulation Time; Parameter (7,6) State Error	147
5-27. Benchmark Controller; Post-Transient Time; Parameter (7,6) State Error	152
5-28. Moving-Bank; Post-Transient Parameter (7,6) State Error	157

	page
5-29 Fixed-Bank; Post-Transient Parameter (7,6) State Error	162
5-30. Fixed-Bank Controller X and U mistuned Post-transient; Parameter point (5,5) . .	169
5-31. Moving-Bank Controller X and U mistuned Post-transient; Parameter point (5,5) . .	174
5-32. Moving-Bank Controller X and U tuned Post-transient; Parameter point (5,5) . .	181
5-33. Fixed-Bank Controller X and U tuned Post-transient; Parameter point (5,5) . .	186
5-34. Benchmark Controller X and U tuned Post-transient; Parameter point (5,5) . .	191

List of Tables

Table	page
3-1. Structural Member's Cross-Sectional Areas	56
3-2. Eigenvalues and Frequencies	68
5-1. Dynamics and Measurement Noise Strengths .	94
5-2. Time Averaged State Estimation Errors Statistics (Q and R Values Equal to the Q and R Values in Table 5-1)	95
5-3. Time Averaged State Estimation Errors Statistics (Q and R Values Greater Than the Q and R Values in Table 5-1)	95
5-4. Dynamics and Measurement Noise Strength Maxium Values for Acceptable Moving Bank Performance	132
5-5. Comparison of Eigenvalues of State Weighting Matrix, Control Weighting Matrix, and Cross Term Matrix	142
5-6. State Weighting and Control Weighting Matrices (7,6)	143
5-7. State Weighting and Control Weighting Matrices (5,5)	180

Abstract

/

This investigation focused on the use of moving-bank multiple model adaptive estimation and control (MMAE). Moving-bank MMAE reduces the computational burden of MMAE by implementing only a subset of the Kalman filters (9 filters versus 100 in this research) that are necessary to describe the system to be estimated/controlled. Important to the development of the moving-bank MMAE are the decision logics governing the selection of the subset of filters. The decision logics cover three situations: initial acquisition of unknown parameter values; tracking unknown parameter values; and reacquisition of the unknown parameters following a "jump" change in these parameter values.

This thesis applied moving-bank MMAE to a rotating two bay truss model of a flexible spacestructure. The rotating two bay truss approximated a space structure that had a hub with appendages extending from the structure. The mass of the hub was large relative to the mass of the appendage. The mathematical model was developed using finite element analysis, transformed into modal formulation, and reduced using a method referred to as singular perturbations.

Multiple models were developed by assuming that variation occurred in the mass and stiffness of the structure.

Results showed that the primary cause for the increase in the performance of the moving-bank algorithm over past efforts was the choice of the measurement covariance noise matrix. This research effort also showed that the space discretization level can play an important role in the performance of the moving-bank MMAC; therefore, care must be taken in determining the discretization level. Finally, this research effort showed that the moving-bank MMAC provided increased control performance over a fixed-bank controller with a discretization that covered the entire range of parameter variation. In addition, the moving-bank controller converged to the performance of a benchmark controller.

MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION
APPLIED TO FLEXIBLE SPACESTRUCTURE CONTROL

1. Introduction

A problem exists in some applications of estimation and control which is caused by the uncertainty of parameters in mathematical models. These uncertain parameters reduce the accuracy which can be expected from the system model and the algorithms based on that model. Their values can be constant yet unknown, slowly varying, or changing abruptly. These changes in parameters often necessitate the identification of parameters within the mathematical model and the changing of the mathematical model during a real-time control problem. This is often referred to as adaptive estimation and control. One important approach to this class of problems is multiple model adaptive estimation (MMAE) and control (MMAC). The specific models in the multiple model structure can either be static ("fixed-bank" MMAE) or dynamically redeclared in real time ("moving-bank" MMAE). This research concentrates on two areas. First, it explores the cause of the moving-bank MMAE performance degradation seen in Karnick's [9] research. In his research, the moving-bank's estimate of the true parameter wandered nonsystematically throughout parameter space.

Second, depending on the cause of the performance degradation, to fine tune either the moving-bank MMAE or the fixed-bank MMAE for best possible performance in a bending spacestructure control application.

1.1. Background

In a system adequately represented by a linear stochastic state model, a full bank MMAE algorithm can be used to alleviate the problem of uncertain parameters. This method consists of designing a Kalman filter for each possible parameter value. A Kalman filter is a recursive data processing algorithm which determines the optimal state estimate for a linear system with dynamics and measurement noises modeled as white and Gaussian. It is further assumed that the uncertain parameters can take on only discrete values; either this is reasonable physically or discrete values are chosen from the continuous parameter variation range. The MMAE approach results in a bank of K Kalman filters, where K is the number of possible parameter values (see Figure 1-1). For each Kalman filter, a conditional probability is produced that assumes its parameter is "correct", given the measurement history. These conditional probabilities are based on the characteristics of the residuals of each Kalman filter and are used as a weighting factor for the state estimate produced by each

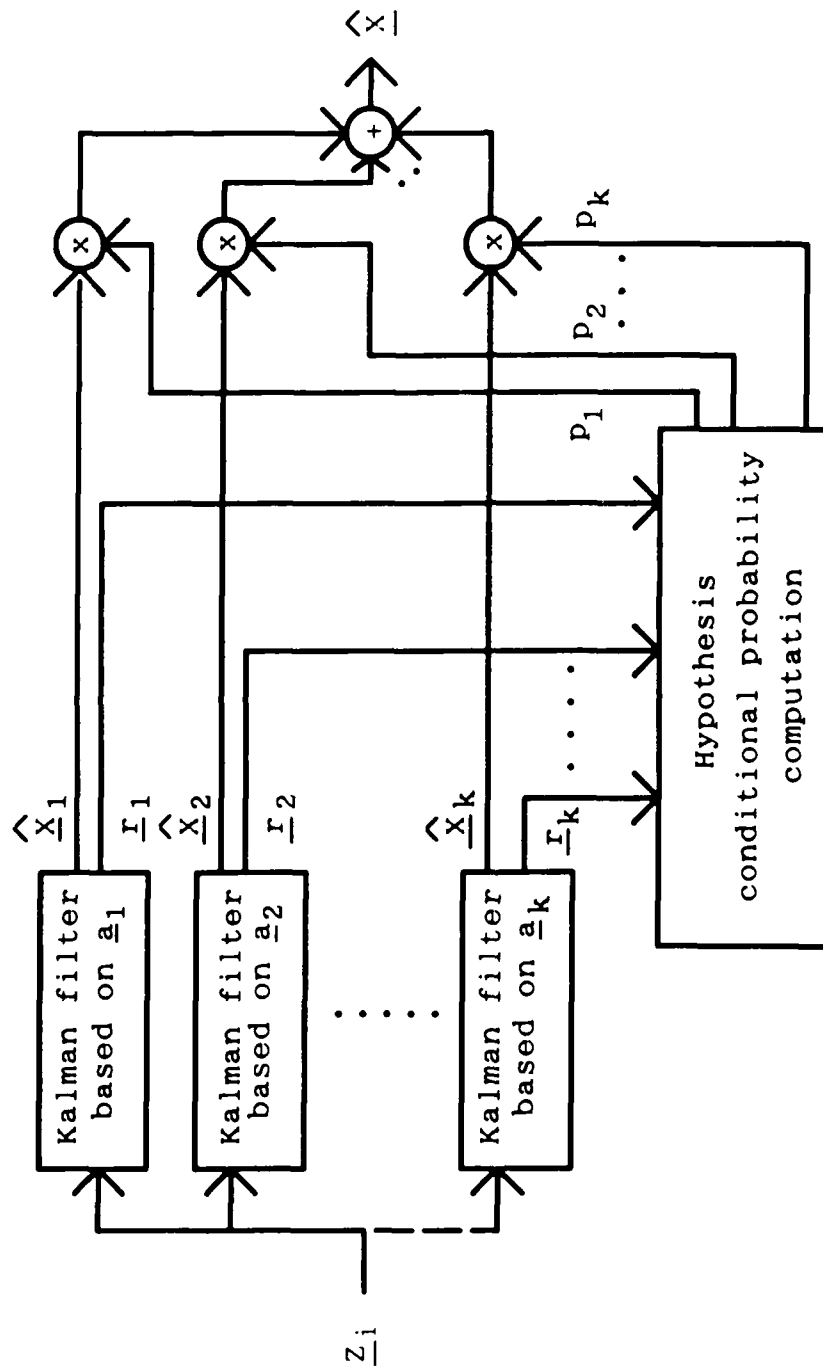


Fig. 1-1. Multiple Model Adaptive Estimator

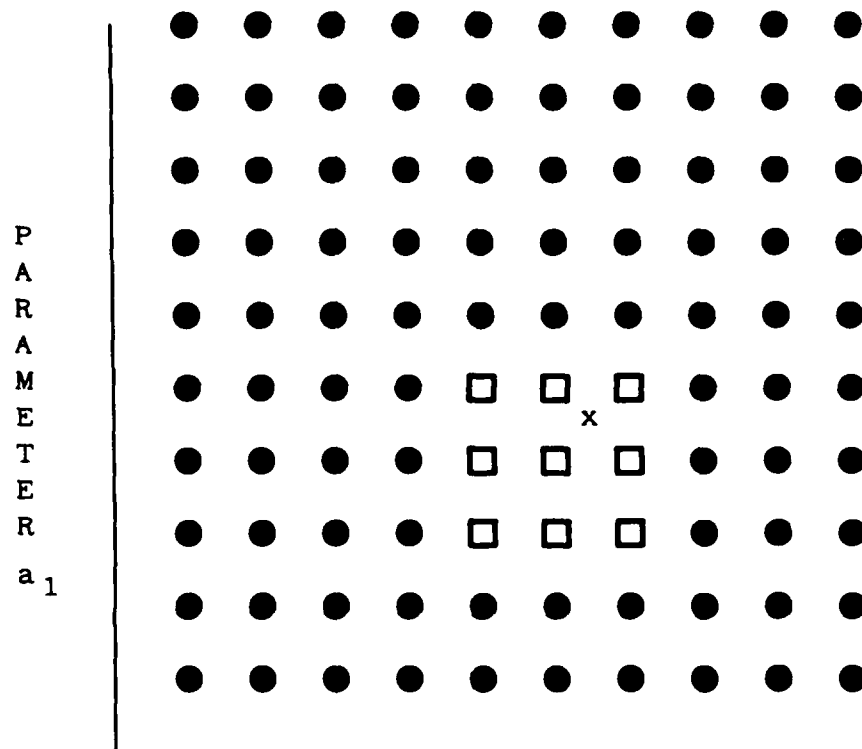
Kalman filter. The Kalman filter based on the best estimate of the "true" parameter, should have residuals that most nearly match the anticipated characteristics of the residuals (as the result of the filter-computed residual covariance matrix) and should have a large conditional probability. In contrast, the filters that correspond to parameter values that are bad estimates of the "true" parameter should have residuals that are larger than anticipated, and they should have smaller conditional probabilities. The adaptive state estimate is then determined by either a probabilistically weighted average of the outputs of each Kalman filter, as seen in Figure 1-1 (this is called the Bayesian form of the MMAE estimator), or by setting the overall state estimate equal to the state estimate of the filter with the highest conditional probability (called the maximum a posteriori, or MAP estimator) [20:365-369].

MMAE has been successfully implemented in several estimation and control problems. The use of a MMAE algorithm to track airborne targets has been researched many times. Three examples of this research are given in references: [10], [16], and [29]. The MMAC has also been used in controlling fuel tank fires [36], addressing terrain correlation [30], and generating estimators for problems in which large initial uncertainties cause non-adaptive extended Kalman filters to diverge [28].

However, one basic problem of the full bank MMAE algorithm is the number of Kalman filters which must be computed simultaneously. For example, if a system had two uncertain parameters and if each of these parameters can take on ten different values, then 100 Kalman filters need to be implemented. As can well be imagined, this is an extremely high computational burden [19:1876].

Several approaches have been used to alleviate the computational burden of a MMAE [3:5]. One method uses Markov processes to model the parameter variation [1;25]. A process is considered Markov if its present parameter value depends only on the single previous parameter value [1:418] and not an entire history of values. Other methods include: using "pruning" and/or "merging" of "decision trees" of the possible parameter time history [24;25], hierarchically structuring the algorithms to reduce the number of filters [4], and a method in which the filter is initialized with a coarse parameter space discretization, but after the filter converges to the "nearest" parameter, the filter is rediscrctized using a simple direct method [15].

To reduce this burden for the example of 2 parameters with 10 possible values for each, a moving bank of Kalman filters consisting of only nine filters can be developed (see Figure 1-2). These Kalman filters should be



- used Kalman filter
- unused Kalman filter
- x current best estimate of the true parameter value

Fig. 1-2. Moving-Bank Multiple Model Adaptive Estimator

associated with the parameter values that most closely surround the estimated "true" parameter value. When the "true" parameter value lies within the region of the bank of Kalman filters, the moving-bank MMAE algorithm behaves essentially the same as the full bank MMAE algorithm. However when the "true" parameter value lies outside of the bank or even inside the bank but near its boundary, the bank must be moved or expanded (see Figure 1-3). A bank is moved or expanded by changing the parameter points to which the nine Kalman filters in the bank correspond [19:1876].

In the past three years, research has been conducted in the area of moving-bank MMAE algorithms to reduce the amount of computational loading needed to estimate both the state variables and uncertain parameters associated with the system model. Initial research conducted by Hentz and Maybeck [6;19] showed that, for a simple but physically motivated system, the moving-bank MMAE algorithm performed as well as the full-bank MMAE algorithm, but with an order of magnitude less computational loading. They also showed that the system performance could be enhanced by starting the bank with a large discretization to allow for initial parameter acquisition and reducing the discretization as the accuracy of the parameter estimate increases. In the follow-on research effort conducted by Fillos [3], he showed that the uncertain parameters associated with the

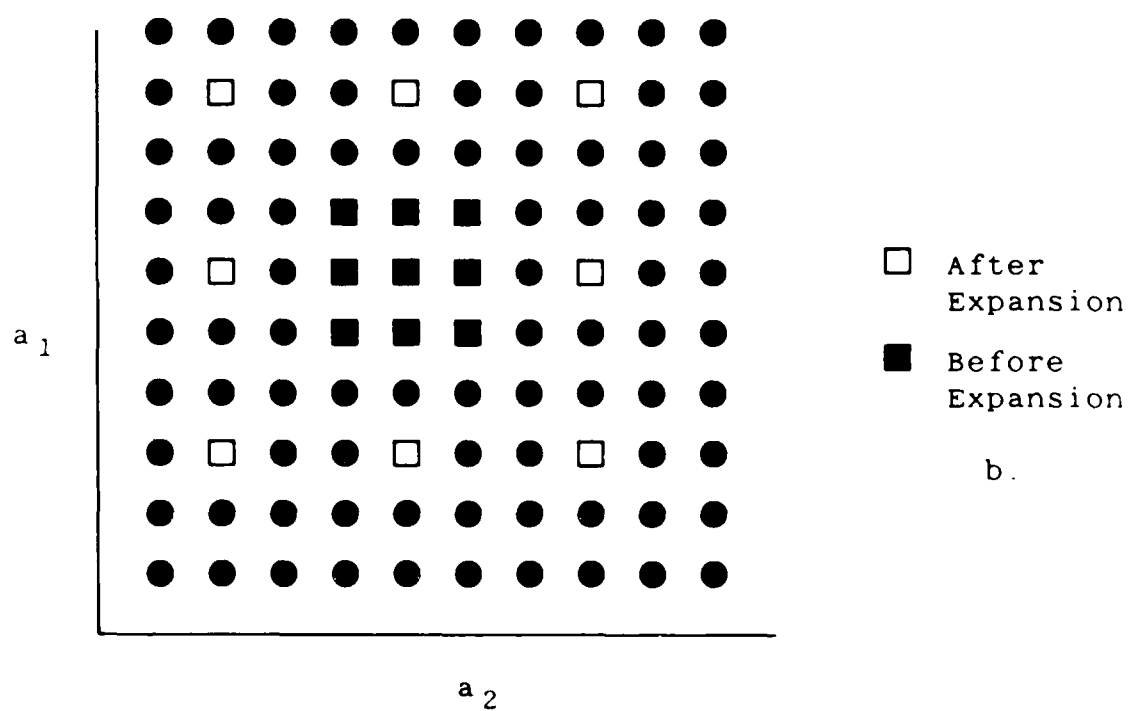
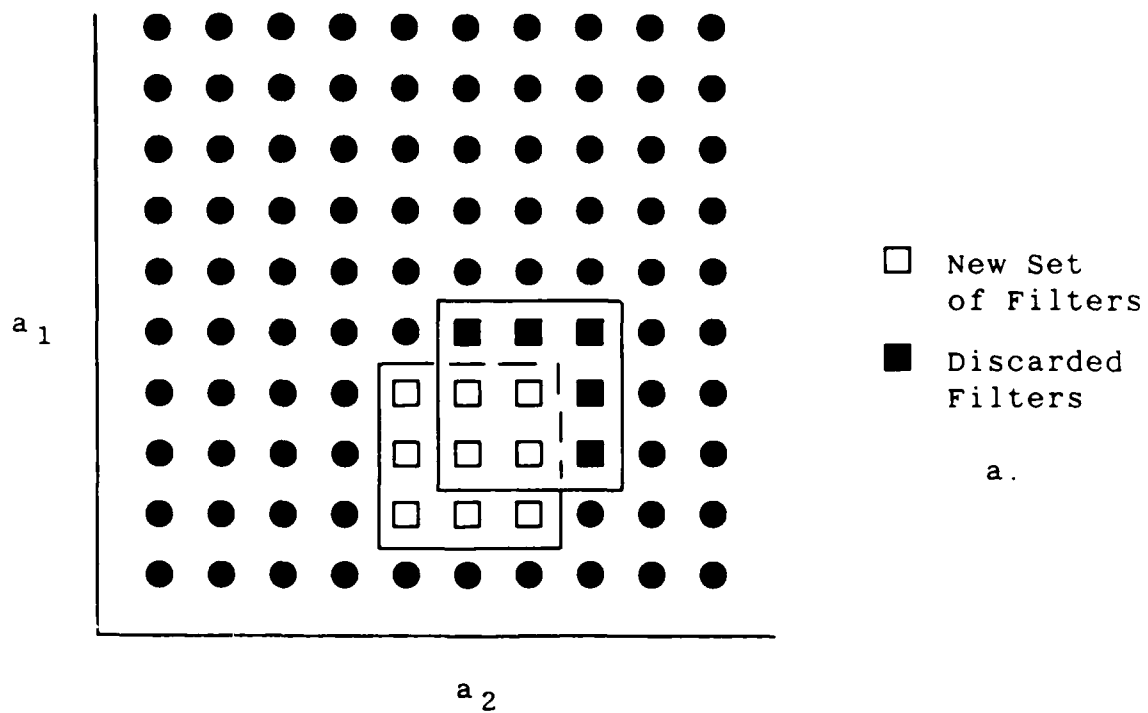


Fig. 1.3. Bank Changes: a. Move b. Expansion

system model can be determined using ambiguity function analysis. The research conducted last year by Karnick and Maybeck showed that a large discretized fixed-bank MMAE algorithm performed at least as well as the moving-bank MMAE algorithm for a more complex system [9]. Furthermore, the moving bank never converged to a consistent parameter value, instead the moving bank wandered nonsystematically throughout the parameter space. The fixed-bank MMAE algorithm included nine Kalman filters similar to the moving-bank MMAE algorithm; however, it used coarser discretization and was not allowed to move [9].

1.2. Problem

The full bank MMAE algorithm is computationally too burdensome for most applications. The moving-bank MMAE was shown to be successful for a physically motivated but simple system [6]. However, the moving-bank MMAE algorithm was shown to have not performed as well as a fixed-bank MMAE algorithm for a more complex space structure application that required adaptive estimation and control [9].

1.3. Scope

This research applies to problems associated with large space structures. The large space structure model used in this research was a two-bay truss (developed by

Drew Karnick in his thesis [9]) attached to a hub (see Figure 3-2) which allowed motion in two degrees of freedom (x-y plane), but did not allow translational motion of the hub. The truss was 100 inches long and 18 inches high. In addition, non-structural masses were added to the truss model for two purposes. First, they allow time varying mass problems to be modeled, such as the mass associated with fuel tanks being depleted. Second, if the nonstructural masses were large, relative to the structure mass, they produce the low frequency structural model associated with large space structures. The space structure model was described in terms of mass and stiffness matrices obtained from a finite element analysis.

Two uncertain parameters were investigated, the non-structural mass and the stiffness matrix within a linear system model structure. These parameters were chosen because they physically appear in the system dynamics equation. However, "nonphysical" parameters could be used if found to produce better estimates. The dynamics and measurement noise characteristics were assumed known and modeled as white Gaussian processes.

1.4. Approach

This research concentrated on two areas. The first was to determine the cause of the moving-bank MMAE

performance degradation seen in Karnick's research and the second was to fine tune either the moving-bank MMAE or the fixed-bank MMAE, depending on the cause of the performance degradation. In previous thesis efforts, Hentz [6], Filios [3], and Karnick [9] developed extensive software programs which were also used in this research. This software included (1) a preprocessor, which created a parameter space that was utilized in a Monte Carlo simulation and ambiguity function analysis (to be discussed at length subsequently), (2) a processor, which simulated the moving-bank or fixed-bank multiple model adaptive estimator, performed Monte Carlo simulation runs and generated data for each run, (3) a postprocessor, which computed the means and variances of variables of interest and then generated the plots of statistics for the Monte Carlo simulation, and (4) a program that computed the ambiguity functions and generated their plots.

Karnick showed that, although the use of a moving bank may provide enhanced state estimation performance, similar performance could have been obtained from a fixed bank estimator with a coarse space discretization that covered the range of parameter variation. The problem that the moving bank had with parameter estimation could have been caused by the dynamic driving noise (Q) and the measurement noise (R) values, system complexity, or unmodelled effects.

Large Q and R values could cause the estimation difficulties by masking the effects of poor assumed parameter values on residuals produced by each filter. The first step in addressing this issue was to select physically realistic measurement devices, e.g., position and velocity measurements or angle measurements. This was accomplished mainly through interviews with Dr. V. B. Venkayya of the Analysis and Optimization Branch, Structural Mechanics Division, Air Force Flight Dynamics Laboratory [34].

Once the measurement devices were chosen, then a sensitivity study was conducted, where the Q and R values in both the filter and truth models were varied simultaneously. This study was conducted because of the natural upper bound on the R and Q values as explained in the preceding paragraph. In conducting this study, the initial Q and R values used were the values Karnick [9] used in his research. The sensitivity study was conducted by initially holding the Q values constant and varying the R values one at a time, where the Q and R matrices were assumed to be diagonal. The R values were decreased until the RMS error for the state estimate no longer decreased substantially. After the optimal R values were determined, the Q values were tuned using the same technique except this time the R values were held constant and the Q values are varied one at a time.

After the appropriate Q and R values were determined, then the first bending mode was extracted from the system model and the truth model. This was done so that an attempt could be made to duplicate Hentz's work [6] for the large space structure system model. Hentz showed that, for a two-state (i.e. one bending mode) filter model matched against a two-state truth model, the center of the moving bank would move in the direction of the true parameter. The moving-bank MMAE algorithm did duplicate Hentz's work and that performance did not degrade substantially when the moving-bank MMAE algorithm was applied to the entire space structure model (where the space structure model and the truth both have a rigid mode and two bending modes). This suggests that the benefits gained with the moving-bank MMAE algorithm do not degrade as the system complexity increases.

Finally, the issue of robustness in the face of unmodelled effects could be addressed by augmenting states to the truth model and then reevaluating the moving-bank MMAE, based on the three-bending-mode space structure model, against the new truth model. By examining the eigenvalues and frequencies of the space structure model before state reduction was performed (see Table 3-2), a truth model with 16 states would be recommended. However, this issue was not addressed in this effort. Since the problem with the moving bank was due to large Q and R

values, the decision was made to conduct the remaining research using a moving-bank MMAE with sufficiently small Q and R values.

Following this decision, the space discretization which produced the best state estimate was determined. An appropriate space discretization level was needed so that the moving-bank algorithm could distinguish between incremental steps in parameter space. In previous research efforts, the space discretization was chosen by intuition. However, investigation was needed into the effect of various discretizations of the parameter space. Insight into the parameter space discretization could be acquired from three sources. The first source of insight could be ambiguity function plots [22;3;9]. Slowly varying ambiguity function plots versus parameter variations, i.e. flat, illustrate that the parameter value associated with the plots was difficult to estimate, but they also illustrate that the parameter has a small effect on the state estimate and therefore did not need to be estimated. In contrast, quickly varying ambiguity function plots, i.e., containing a sharp peak and valleys, illustrate that the parameter could be estimated and does substantially affect the state estimate. The faster the ambiguity function plots varied, the tighter the space discretization needs to be. However, Karnick [9] experienced extreme difficulties in analysing

the ambiguity function plots because they consisted of multiple peaks. The multiple peaks made determining the size of the space discretization level difficult if not impossible. The second source of insight could be the values of the dynamics driving noise strength (Q) and measurement noise covariance (R) [24]. Large Q and R values indicated a need for large space discretization because such large Q and R values would mask differences in the residuals of filters in a tightly spaced discretization. This difference in residuals was critical for the operation of a MMAE. The third source of insight could be a sensitivity study conducted by giving the filter an incorrect parameter value and then observing the state estimate [9]. The more inaccurate the state estimate becomes, for a given size parameter variation, the tighter the space discretization needs to be for that parameter. This technique was used because of the multiple peak problem experienced with the ambiguity function plots.

Following the space discretization study, a sensitivity study was performed to determine the upper bounds on the dynamics driving noise (Q) and measurement covariance noise (R). These bounds determine the precision the Q and R values must have in order for the moving-bank algorithm to perform correctly. This study differs from the previous Q and R sensitivity study by determining the actual bound on the precision of the Q and R values, while the previous

study only wanted to choose Q and R values lower than the bound values. In conducting this study, the R values were held constant while the Q values were increased one at a time by orders of magnitude until the adaptive mechanization began to degrade. Then the Q values were held constant while the R values were varied one at a time by orders of magnitudes until the adaptive mechanization began to degrade.

Following the Q and R sensitivity study, a study was conducted on the controller performance sensitivity to altered cost function definition. In the research Karnick performed, the quadratic cost function used to determine the full-state feedback gain (\underline{G}_C^*) had no cross terms. However, if control influence is desired on the states over the entire sample period then cross terms can be non-zero. Therefore, in this study, the cross weighting matrix (S) was evaluated [23:76] and compared to the state weighting matrix (X) and the control weighting matrix (U). The comparison showed that the appropriately chosen cross weighting matrix was small compared to the X and U matrices. This was mostly due to the very small sample period time as compared to the system characteristic times. Therefore, the cross weighting matrix was not added to the quadratic cost function.

Following this decision, the X and U matrices were tuned to provide for small rms values on the true states

while not driving the control inputs beyond their maximum value. The initial X and U matrices were equal to the X and U matrices used by Karnick. The tuning process was accomplished by holding the U values constant while varying the X values one at a time, then the X values were held constant while varying the U values one at a time.

After the X and U values were tuned, a study was conducted on transitioning from nonadaptive controller gains to adaptive controller gains based on parameter identification. This study was conducted because if adaptive control inputs were supplied to the system before an accurate estimate of the true parameter was obtained, the control inputs could drive the system unstable. Therefore, a nonadaptive controller was designed that was very robust to parameter variation at the expense of controller performance so that the controller inputs would at least maintain stability regardless of the estimate of the true parameter. However, the moving bank's estimate of the true parameter was locking onto the actual true parameter so quickly that the nonadaptive controller was not required. Therefore, there was no transition from nonadaptive control to adaptive control in this research.

The final study was a comparison between three controllers (a moving-bank controller, a fixed-bank controller, and a benchmark controller). Ideal performance

was created using a benchmark controller. In the benchmark controller only one filter was matched against the truth model and this filter was given artificial knowledge of the true parameter. The fixed-bank controller was designed identically to the moving-bank controller except the nine implemented filters were not allowed to move in parameter space. The performance criteria used in the controller study was the mean and standard deviation on the true states.

1.5 Summary

The remaining chapters in this thesis cover the following areas. Chapter II develops the algorithms for the moving-bank MMAE and MMAC. Chapter III develops the large spacestructure model. Chapter IV describes the simulations used in this thesis and gives a brief description of how the simulations are to be used in the research. Chapter V details the results of the research. Finally, Chapter VI contains the conclusions and recommendations.

II. Algorithm Development

2.1. Introduction

This chapter develops the algorithms for the fullscale and moving-bank Bayesian Multiple Model Adaptive Estimator. First, the full-scale model is developed. This is then modified for the moving bank case. Finally, the Ambiguity Function analysis is developed.

2.2. Bayesian Multiple Model Adaptive Estimation Algorithm Development

Development of the full-scale Bayesian Multiple Model Adaptive Estimation (MMAE) algorithms is presented in this section. For a more rigorous development, the reader is directed to reference [22:129-136].

Let the system under consideration be discrete and described by [3; 6; 21; 9]:

$$\underline{x}(t_{i+1}) = \Phi(t_{i+1}, t_i) \underline{x}(t_i) + B_d(t_i) \underline{u}(t_i) + G_d(t_i) \underline{w}_d(t_i)$$

$$\underline{z}(t_i) = H(t_i) \underline{x}(t_i) + \underline{v}(t_i) \quad (2-1)$$

where " $\underline{\cdot}$ " denotes a vector stochastic process and:

$$\underline{x}(t_i) = n\text{-dimensional state vector}$$

$\Phi(t_{i+1}, t_i)$ = state transition matrix

$\underline{u}(t_i)$ = r-dimensional known input vector

$B_d(t_i)$ = control input matrix

$\underline{w}_d(t_i)$ = s-dimensional white Gaussian dynamics
noise vector

$G_d(t_i)$ = noise input matrix

$\underline{z}(t_i)$ = m-dimensional measurement vector

$H(t_i)$ = measurement matrix

$\underline{v}(t_i)$ = m-dimensional white Gaussian
measurement noise vector

and the following statistics apply:

$$E[\underline{w}_d(t_i)] = \underline{0}$$

$$E[\underline{w}_d(t_i) \underline{w}_d^T(t_j)] = Q_d(t_i) \delta_{ij},$$

$$E[\underline{v}(t_i)] = \underline{0}$$

$$E[\underline{v}(t_i) \underline{v}^T(t_j)] = R(t_i) \delta_{ij},$$

where δ is the Kronecker delta function. It is also assumed that $\underline{x}(t_0)$, $\underline{w}_d(t_i)$, and $\underline{v}(t_i)$ are independent for all times t_i .

All of the following matrices within Equation (2-1) are functions of system parameters: Φ , B_d , G_d , Q_d , H , and R . Now suppose these system parameters have values that are not known with absolute certainty or that can vary. To address this problem, let \underline{a} be the uncertain p-dimensional parameter vector which is an element of A , where A is a

subset of R^P (in this case R stands for the set of real numbers and R^P is real Euclidean p -dimensional space). This parameter vector may be uncertain but constant, it may be slowly varying, or it may undergo jump changes. The purpose of a Bayesian estimator conceptually is to compute the following conditional density function [22]:

$$f_{\underline{x}(t_1), \underline{a} | \underline{z}(t_1)}(\underline{x}, \underline{a} | \underline{z}_1) = f_{\underline{x}(t_1) | \underline{a}, \underline{z}(t_1)}(\underline{x} | \underline{a}, \underline{z}_1) * f_{\underline{a} | \underline{z}(t_1)}(\underline{a} | \underline{z}_1) \quad (2-2)$$

where $\underline{z}(t_1)$ is the vector of measurements from t_0 to t_1 ,

$$\underline{z}(t_1) = [\underline{z}^T(t_1), \underline{z}^T(t_{1-1}), \dots, \underline{z}^T(t_0)]^T$$

The second term on the right side of Equation (2-2) can be further evaluated [22]:

$$\begin{aligned} f_{\underline{a} | \underline{z}(t_1)}(\underline{a} | \underline{z}_1) &= f_{\underline{a} | \underline{z}(t_1), \underline{z}(t_{1-1})}(\underline{a} | \underline{z}_1, \underline{z}_{1-1}) \\ &= \frac{f_{\underline{a}, \underline{z}(t_1) | \underline{z}(t_{1-1})}(\underline{a}, \underline{z}_1 | \underline{z}_{1-1})}{f_{\underline{z}(t_1) | \underline{z}(t_{1-1})}(\underline{z}_1 | \underline{z}_{1-1})} \\ &= \frac{f_{\underline{z}(t_1) | \underline{a}, \underline{z}(t_{1-1})}(\underline{z}_1 | \underline{a}, \underline{z}_{1-1}) f_{\underline{a} | \underline{z}(t_{1-1})}(\underline{a} | \underline{z}_{1-1})}{\int_A f_{\underline{z}(t_1) | \underline{a}, \underline{z}(t_{1-1})}(\underline{z}_1 | \underline{a}, \underline{z}_{1-1}) f_{\underline{a} | \underline{z}(t_{1-1})}(\underline{a} | \underline{z}_{1-1}) d\underline{a}} \quad (2-3) \end{aligned}$$

Conceptually, Equation (2-3) can be solved recursively, starting from an a priori probability density function of \underline{a} , since $f_{\underline{z}(t_1)|\underline{a},\underline{z}(t_{1-1})}(\underline{z}_1|\underline{a},\underline{z}_{1-1})$ is Gaussian with a mean of $H(t_1)\hat{\underline{x}}(t_1^-)$ and covariance $[H(t_1)P(t_1^-)H^T(t_1)+R(t_1)]$, where $\hat{\underline{x}}(t_1^-)$ and $P(t_1^-)$ are the conditional mean and covariance respectively of $\underline{x}(t_1)$ just prior to the measurement at t_1 , assuming a particular realization \underline{a} of \underline{a} [22].

Using the conditional mean, the estimate of $\underline{x}(t_1)$ becomes [22]:

$$\begin{aligned} E[\underline{x}(t_1)|\underline{z}(t_1) = \underline{z}_1] &= \int_{-\infty}^{\infty} \underline{x} \cdot f_{\underline{x}(t_1)|\underline{z}(t_1)}(\underline{x}|\underline{z}_1) d\underline{x} \\ &= \int_{-\infty}^{\infty} \underline{x} \cdot \left[\int_A f_{\underline{x}(t_1),\underline{a}|\underline{z}(t_1)}(\underline{x},\underline{a}|\underline{z}_1) d\underline{a} \right] d\underline{x} \end{aligned} \quad (2-4)$$

$$\begin{aligned} E[\underline{x}(t_1)|\underline{z}(t_1) = \underline{z}_1] &= \int_{-\infty}^{\infty} \underline{x} \left[\int_A f_{\underline{x}(t_1)|\underline{a},\underline{z}(t_1)}(\underline{x}|\underline{a},\underline{z}_1) \right. \\ &\quad \left. f_{\underline{a}|\underline{z}(t_1)}(\underline{a}|\underline{z}_1) d\underline{a} \right] d\underline{x} \\ &= \int_A \left[\int_{-\infty}^{\infty} \underline{x} f_{\underline{x}(t_1)|\underline{a},\underline{z}(t_1)}(\underline{x}|\underline{a},\underline{z}_1) d\underline{x} \right] f_{\underline{a}|\underline{z}(t_1)}(\underline{a}|\underline{z}_1) d\underline{a} \end{aligned} \quad (2-5)$$

where the term in brackets is the estimate of $\underline{x}(t_1)$ based on a particular value of the parameter vector. This is the output of the Kalman filter based on that realization of the parameter vector. When \underline{a} is continuous over A , this requires an infinite number of filters in the bank. To reduce the number of filters, the parameter space is

usually discretized, yielding a finite number of filters. The integrals over A in Equations (2-4) and (2-5) then become summations. Defining $p_k(t_1)$ as the probability that the k^{th} elemental filter is correct, conditioned on the measurement history, it can be shown by a method analogous to the development for Equation (2-3) that $p_k(t_1)$ satisfies [22]:

$$p_k(t_1) = \frac{f_{\underline{z}}(t_1) | \underline{a}, \underline{z}(t_{1-1}) (\underline{z}_1 | \underline{a}_k, \underline{z}_{1-1}) * p_k(t_{1-1})}{\sum_{j=1}^K f_{\underline{z}}(t_1) | \underline{a}, \underline{z}(t_{1-1}) (\underline{z}_1 | \underline{a}_j, \underline{z}_{1-1}) * p_j(t_{1-1})} \quad (2-6)$$

$$\hat{\underline{x}}(t_1) = E[\underline{x}(t_1) | \underline{z}(t_1) = \underline{z}_1] = \sum_{j=1}^K \hat{\underline{x}}_k(t_1^+) * p_k(t_1) \quad (2-7)$$

where $\underline{a} \in [a_1, a_2, \dots, a_K]$ and $\hat{\underline{x}}_k(t_1^+)$ is the mean of $\underline{x}(t_1)$ conditioned on $\underline{a} = \underline{a}_k$ and $\underline{z}(t_1) = \underline{z}_1$, i.e. the output of the k^{th} Kalman filter in the bank, based on the assumption $\underline{a} = \underline{a}_k$. Pictorially, the algorithm appears as in Figure 1-1.

The probability weighting factors for each Kalman filter are calculated from Equation (2-6) [22], where

$$\begin{aligned} & f_{\underline{z}}(t_1) | \underline{a}, \underline{z}(t_{1-1}) (\underline{z}_1 | \underline{a}_k, \underline{z}_{1-1}) \\ &= 1 / \{ (2\pi)^{m/2} |A_k(t_1)|^{1/2} \} \exp[-(1/2) \underline{z}_k^T(t_1) A_k^{-1}(t_1) \underline{z}_k(t_1)] \end{aligned} \quad (2-8)$$

and

$$A_k(t_1) = H_k(t_1)P_k(t_1)H_k^T(t_1) + R_k(t_1)$$

$$\underline{r}_k(t_1) = \underline{z}_1 - H_k(t_1)\hat{\underline{x}}_k(t_1)$$

m = number of measurements

Both the residual covariance $A_k(t_1)$ and the residual $\underline{r}_k(t_1)$ itself are readily available from the k^{th} elemental filter. The estimate of the parameter and the covariance of the parameter are given by [22]:

$$\begin{aligned}\hat{\underline{a}}(t_1) &\triangleq E[\underline{a} | \underline{z}(t_1) = \underline{z}_1] = \int_{-\infty}^{\infty} \underline{a} f_{\underline{a} | \underline{z}(t_1)}(\underline{a} | \underline{z}_1) d\underline{a} \\ &= \int_{-\infty}^{\infty} \underline{a} \left[\sum_{k=1}^K p_k(t_1) \delta(\underline{a} - \underline{a}_k) \right] d\underline{a} \\ &= \sum_{k=1}^K \underline{a}_k p_k(t_1)\end{aligned}\quad (2-9)$$

and

$$\begin{aligned}E\{[\underline{a} - \hat{\underline{a}}(t_1)][\underline{a} - \hat{\underline{a}}(t_1)]^T | \underline{z}(t_1) = \underline{z}_1\} \\ = \sum_{k=1}^K [\underline{a}_k - \hat{\underline{a}}(t_1)][\underline{a}_k - \hat{\underline{a}}(t_1)]^T p_k(t_1)\end{aligned}\quad (2-10)$$

The covariance of the state estimate is given by [22]:

$$\begin{aligned}P(t_1) &= E\{[\underline{x}(t_1) - \hat{\underline{x}}(t_1)][\underline{x}(t_1) - \hat{\underline{x}}(t_1)]^T | \underline{z}(t_1) = \underline{z}_1\} \\ &= \int_{-\infty}^{\infty} [\underline{x} - \hat{\underline{x}}(t_1)][\underline{x} - \hat{\underline{x}}(t_1)]^T f_{\underline{x}(t_1) | \underline{z}(t_1)}(\underline{x} | \underline{z}_1) d\underline{x} \\ &= \sum_{k=1}^K p_k(t_1) \int_{-\infty}^{\infty} [\underline{x} - \hat{\underline{x}}(t_1)][\underline{x} - \hat{\underline{x}}(t_1)]^T \\ &\quad f_{\underline{x}(t_1) | \underline{a}, \underline{z}(t_1)}(\underline{x} | \underline{a}_k, \underline{z}_1) d\underline{x}\end{aligned}$$

$$= \sum_{k=1}^K P_k(t_1) [P_k(t_1^+) + [\hat{\underline{x}}_k(t_1^+) - \hat{\underline{x}}(t_1^+)]^* [\hat{\underline{x}}_k(t_1^+) - \hat{\underline{x}}(t_1^+)]^T] \quad (2-11)$$

where $P_k(t_1^+)$ is the covariance of the state estimate of the k^{th} elemental filter.

The Bayesian MMAE has been shown to be optimal and to converge if the true value of the parameter is nonvarying [5]. Convergence for this case occurs when the probability associated with one elemental filter is essentially one and the probability associated with all other elemental filters is essentially zero. The MMAE converges to the elemental filter with parameter value equal to, or most closely representing, the true parameter set, as defined in [5].

There are no theoretical results available for varying parameters [3:18; 6:8; 9:20]. The fact that the filter can converge to one filter for a non-varying true parameter value, does give reason for some concern. For example, if the true parameter value is varying rather slowly, the algorithm may assume one filter is correct with probability essentially equal to one. However, the true parameter value may eventually become significantly different from the value initially estimated by the filter [6:9], resulting in filter divergence.

Another possibility is that the algorithm may converge and lock onto the "wrong" filter. The filter is, to

some degree, always based on an erroneous model and may converge to the wrong parameter point [22:23]. Dasgupta and Westphal investigated the case of unknown biases in the measurement processes and showed that the algorithm may converge to a parameter point that is not close to the true value of the parameter space [3:17; 6:8].

One method of preventing divergence is to add dynamics pseudonoise to the assumed model [22:25] in each elemental filter; however, too much pseudonoise addition tends to "mask" the difference between the "correct" and "incorrect" filters. The performance of the MMAE is dependent upon significant differences between the residual characteristics of the "correct" versus "incorrect" elemental filters. If the residuals are consistently of the same magnitude, Equations (2-6) and (2-8) show that the filter with the smallest $|A_k|$, will experience an increase in its probability weighting; however, $|A_k|$ is independent of the residuals as well as the "correctness" of the k^{th} model [22:133].

Hentz, Filios, and Karnick [3; 6; 9] prevented the "lock on" problem discussed previously by fixing the lower bound of the probabilities associated with the implemented filters [1; 22:27]. If the computed value of any probability falls below a threshold, it is reset to some minimum value determined by performance analysis, and then all

probabilities are rescaled so that their sum would remain equal to unity.

2.3. Moving-Bank Algorithm Development [3:22-33]

The MMAE generates a computational burden that is too large for most practical applications [3; 7; 19; 9]. Maybeck and Hentz demonstrated that the full bank of filters could be replaced by a subset of filters based on discrete parameter values "closest" to the current estimate of the parameter vector. The probability associated with non-implemented filters is conceptually set to zero while the probability weightings are distributed among the implemented filters. As the parameter set estimate changes, filters that are "closer" to the new parameter estimate are implemented while those "farthest" away are removed. Maybeck and Hentz also investigated changing the discretization levels of the moving bank model. During the acquisition stage, the implemented filters are set to a coarse discretization, then changed to finer discretizations as the parameter estimate improves. Therefore, the implemented filters do not necessarily occupy adjacent discrete points in the parameter space, as would be used in the full bank MMAE.

2.3.1. Weighted Average [3; 6; 9]. The outputs of each elemental filter of the moving bank estimator are

weighted and summed in the same manner as Equations (2-6) and (2-7); however, only the implemented filters in the moving bank are summed. If J filters are implemented, Equation (2-7) becomes:

$$\hat{\underline{x}}(t_i^+) = \sum_{j=1}^J \hat{\underline{x}}_j(t_i^+) p_j(t_i) \quad (2-12)$$

Similarly, Equation (2-6) describing the $p_k(t_i)$'s becomes:

$$p_j(t_i) = \frac{f_{\underline{z}(t_i) | \underline{a}, \underline{z}(t_{i-1})}(\underline{z}_i | \underline{a}_j, \underline{z}_{i-1}) * p_j(t_{i-1})}{\sum_{k=1}^J f_{\underline{z}(t_i) | \underline{a}, \underline{z}(t_{i-1})}(\underline{z}_i | \underline{a}_k, \underline{z}_{i-1}) p_k(t_{i-1})} \quad (2-13)$$

and Equation (2-8) similarly is:

$$\begin{aligned} & f_{\underline{z}(t_i) | \underline{a}, \underline{z}(t_{i-1})}(\underline{z}_i | \underline{a}_j, \underline{z}_{i-1}) \\ &= 1 / \{ (2\pi)^{m/2} |A_j(t_i)|^{1/2} \} \exp \{ -(1/2) \underline{z}_j^T(t_i) A_j^{-1}(t_i) \underline{z}_j(t_i) \} \end{aligned} \quad (2-14)$$

and

$$\begin{aligned} A_j(t_i) &= H_j(t_i) P_j(t_{i-1}) H_j^T(t_i) + R_j(t_i) \\ \underline{z}_j(t_i) &= \underline{z}_i - H_j(t_i) \hat{\underline{x}}_j(t_{i-1}) \end{aligned}$$

m = the dimension of \underline{z} (number of measurements)

R_j = the measurement noise covariance in the j^{th} elemental filter.

2.3.2. Sliding the Moving Bank [3:25]. The moving bank MMAE is a smaller version of the full-bank MMAE, with the moving bank centered around a parameter estimate. Typically, the moving bank is not initially centered on the true parameter point, and the true parameter point may change. This makes the decision logic for moving the "bank" of critical importance. Several algorithms have previously been investigated for this purpose, including Residual Monitoring, Parameter Position Estimate Monitoring, Parameter Position and Velocity Estimate Monitoring, and Probability Monitoring [3; 6; 19; 9].

2.3.2.1. Residual Monitoring. Let a likelihood quotient for each elemental filter, $L_j(t_1)$, be defined as the quadratic form appearing in Equation (2-8):

$$L_j(t_1) = \underline{r}_j^T(t_1) A_j^{-1}(t_1) \underline{r}_j(t_1) \quad (2-15)$$

The decision is made to move the bank if at time t_1 :

$$\min[L_1(t_1), L_2(t_1), \dots, L_j(t_1)] \geq T \quad (2-16)$$

where T is a threshold level with a numerical value that is determined during performance evaluations. If the true parameter vector value is outside the moving bank, it would be expected that all the likelihood quotients exceed the threshold. The bank is then moved in the direction of parameter space corresponding to the filter with the smallest L_j , as that filter would be expected to be nearest to the true parameter set. This method should respond quickly to a real need to move the bank but also gives erroneous results for a single instance of large residuals, possibly due to noise corruption.

2.3.2.2. Probability Monitoring. This method is similar to residual monitoring except that the conditional hypothesis probabilities, generated by Equation (2-6), are monitored. If the conditional hypothesis probability associated with an elemental filter is larger than a previously determined threshold, the bank is centered on that filter. Maybeck and Hentz found this decision logic, as well as parameter position monitoring, to provide the best performance. However, probability monitoring required fewer computations than parameter position monitoring [6:93-99]

2.3.2.3. Parameter Position Estimate Monitoring. This method centers the bank around the current estimate

of the true parameter set, which is given by:

$$\hat{a}(t_i) = \sum_{j=1}^J a_j p_j(t_i) \quad (2-17)$$

where J is the number of filters implemented in the moving bank. Movement is initiated when the bank is not centered on the point closest to the current true parameter set estimate [3:26].

2.3.2.4. Parameter Position and Velocity

Estimate Monitoring. This method estimates the velocity of the parameter position using the history of parameter position estimates. The velocity estimate is used to estimate the position of the parameter set at the next sample time. The bank is centered at this estimate of the future parameter point, thereby adding "lead" into the positioning of the bank [24]. Maybeck and Hentz found this decision logic performed worse than parameter position estimate monitoring or probability monitoring [6:85; 19:23], not providing much desired lead but causing reduced stability in the bank location.

2.3.3. Bank Contraction and Expansion. The filters in the moving bank model do not necessarily have to be at adjacent discretized parameter values; see Figure 2-1. This may decrease the accuracy of the initial estimate but it increases the probability that the true parameter set lies within the bank, thereby enhancing initial

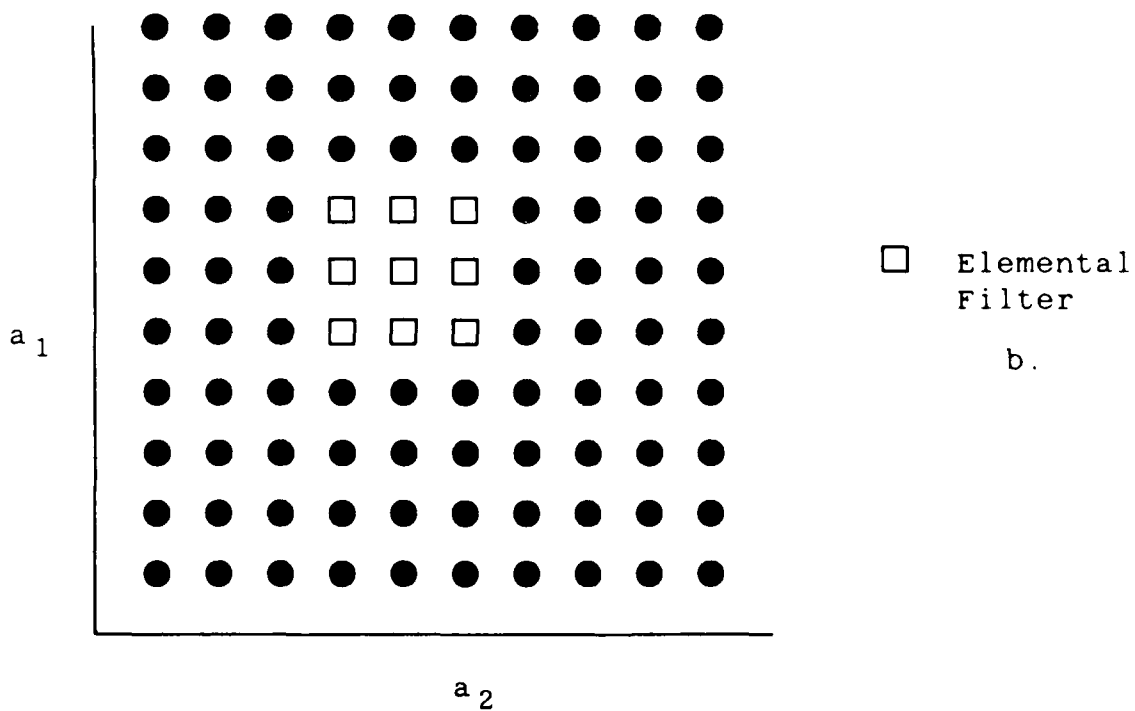
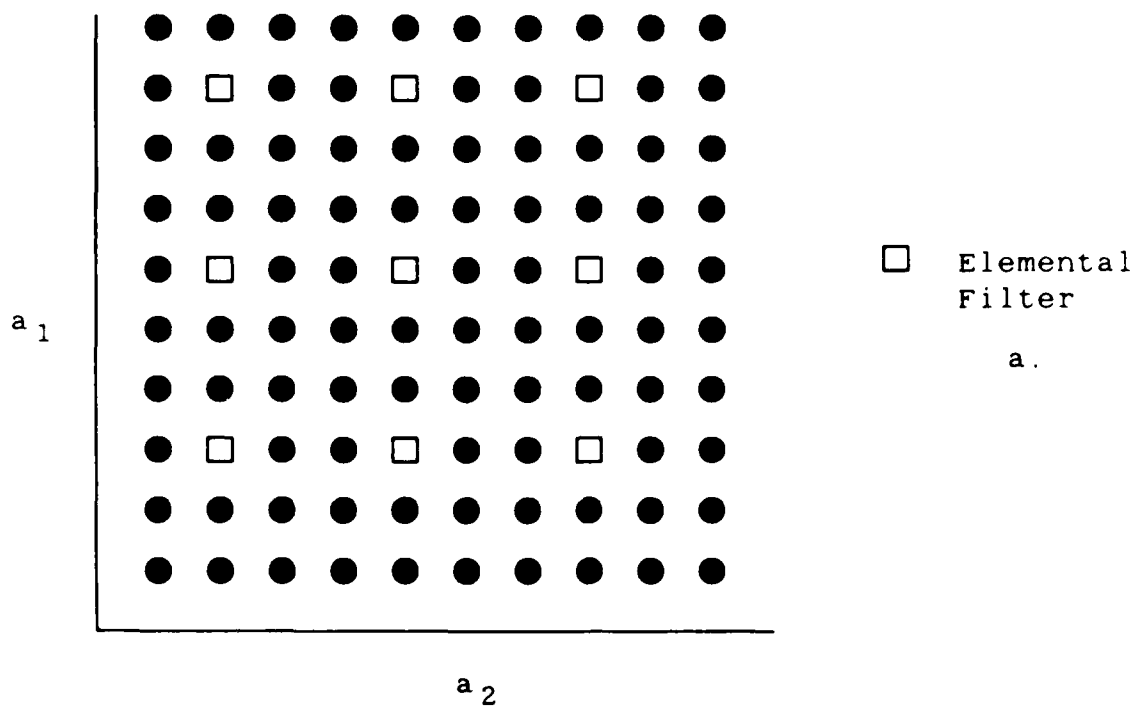


Fig. 2-1. Bank Discretizations: a. Coarse b. Fine

convergence. Maybeck and Hentz found that parameter acquisition performance can be improved by starting the moving bank with a coarse discretization so that the entire parameter value range lies within the bank and then contracting the bank into a finer discretization when the parameter covariance (Equation (2-10)) drops below some selected threshold [3:28; 6:26; 19:25].

Karnick proposed another method [9] that may improve acquisition. This method monitors the probability associated with a "side" of the bank; see Figure 2-2, and would be calculated as:

$$P_{\text{side}}(t_1) = \frac{\sum_{\text{side}} f_j(\underline{z}(t_1))}{\sum_{4 \text{ sides}} f_j(\underline{z}(t_1))} \quad (2-18)$$

where $f_j(\underline{z}(t_1))$ is the probability distribution function of the j^{th} Kalman filter which is determined using Equation (2-14). Several possibilities exist for threshold logic. If the probability associated with a side falls below a certain threshold, it can be moved toward the center of the movable bank. Conversely, if the probability associated with a side rises above some threshold, the remaining three sides are moved toward the center of the movable bank. A third possibility is moving in all four sides if the summed probability of all the "side" filters is below some threshold.

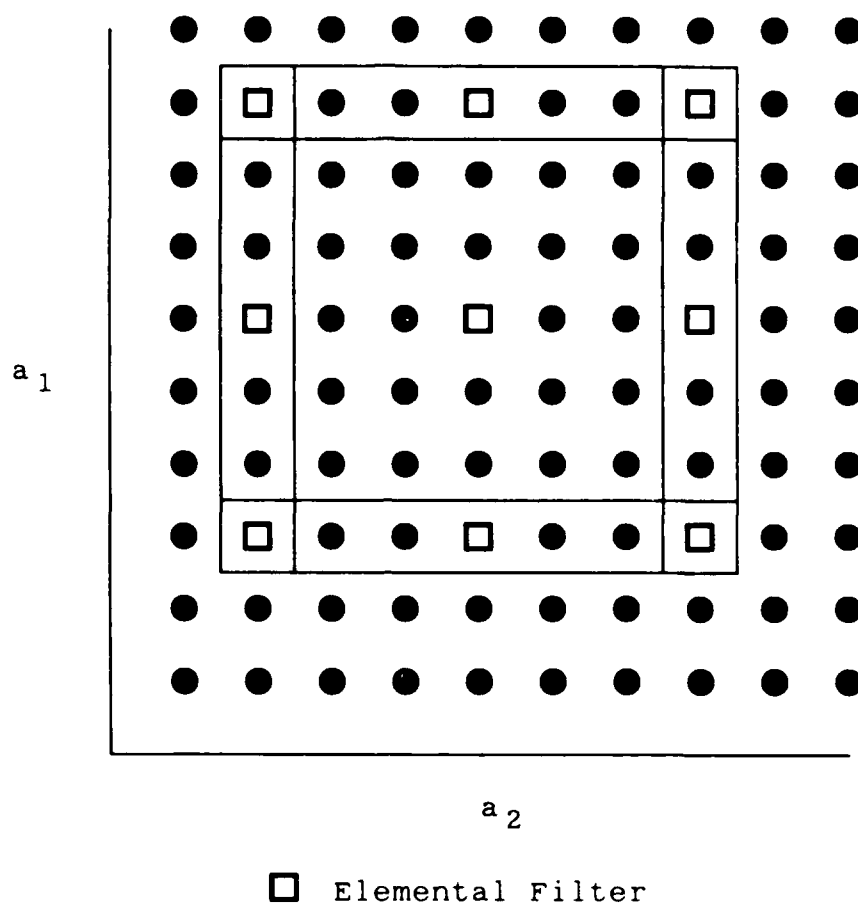


Fig. 2-2. Probability Weighting of Sides

The bank may also need to be expanded if the true parameter value undergoes a jump change to a point outside the range covered by the bank. This jump change can be detected by residual or probability monitoring. For residual monitoring, the likelihood ratios for all the implemented filters are expected to be large and to exceed some threshold. For probability monitoring, it is expected that the conditional hypothesis probabilities be "close" in magnitude; however, performance analysis must be performed to confirm this. The subsequent bank contraction is accomplished in the same manner as discussed in the previous paragraphs.

2.3.4. Initialization of New Elemental Filters

[3:29-31; 6:26-30; 9:29-32]. When the decision is made to move, expand, or contract the bank, new filters must be brought on line and "incorrect filters" discarded. New filters require new values for Φ , B_d , K (steady state Kalman gain matrix), H , $\hat{x}_j(t_1^+)$, and $p_j(t_1)$. Except for the last two terms, these are predetermined values associated with the particular filter being implemented because for this research the system model is assumed to be linear and time invariant.

The current moving bank estimate of $\hat{x}(t_1^+)$ is an appropriate choice for $\hat{x}_j(t_1^+)$ for a new elemental filter. The value for $p_j(t_1)$ is dependent on the number of new

filters being implemented. If the bank "slides," as shown in Figure 1-3a, this involves either three or five new filters. The probability weighting of the discarded filters can be redistributed equally among the new filters or in a manner that indicates the estimated "correctness" of the new filter and still maintains the sum of the probabilities equal to one. Hentz suggested the following [7:29]:

$$p_{jch}(t_1) = \frac{f_j(\underline{z}(t_1))(1 - \sum_{unch} p_k(t_1))}{\sum f_k(\underline{z}(t_1))} \quad (2-19)$$

where ch = changed, unch = unchanged, and where $f_j(\underline{z}(t_1))$ is defined in Equation (2-14) but with the residual replaced by:

$$\underline{r}_j(t_1) = \underline{z}_1 - H_j \hat{\underline{x}}_j(t_1^+) \quad (2-20)$$

However, this requires additional computations and has demonstrated no significant performance improvement over dividing the probability weighting equally among the changed filters [6:104].

A bank expansion or contraction can result in the resetting of all the filters in the bank as shown in

Figure 1-3b. Dividing the probability weighting equally among the new filters is appropriate since the old probability weightings may no longer be valid.

2.4. Controller and Estimator Design [3:18-22; 6:33-43; 9:32-39]

Several controller and estimator designs are appropriate for implementation with the moving bank or full-bank MMAE. All designs considered use the "assumed certainty equivalence design" technique [23:241], which consists of developing an estimator cascaded with a deterministic full-state feedback optimal controller. This method assumes independence between controller and estimator design and is the optimal stochastic controller design for a linear system driven by white Gaussian noise with quadratic performance criterion [23:171].

The moving-bank MMAE or the fixed bank MMAE is the estimator used in this thesis. Each elemental estimator within the full bank is a constant-gain Kalman filter associated with a particular point in the parameter space. Each design assumes a time invariant system with stationary noise. Propagation of the elemental filter estimate, $\hat{\underline{x}}_k(t)$, is given by:

$$\hat{\underline{x}}_k(t_1^-) = \Phi \hat{\underline{x}}_k(t_{1-1}^+) + B_{dk} u(t_{1-1}) \quad (2-21)$$

and the estimate is updated by:

$$\hat{\underline{x}}_k(t_i^+) = \hat{\underline{x}}_k(t_i^-) + K_k[z(t_i) - H_k\hat{\underline{x}}_k(t_i^-)] \quad (2-22)$$

where the subscript "k" indicates association with a particular point \underline{a}_k in the parameter space.

The design of each controller is similar. Each is a linear, quadratic cost, (LQ) full-state feedback optimal deterministic controller, based on a time invariant error state space formulation and constant cost weighting matrices in the quadratic cost to be minimized [21:297]. The form of the cost equation to be minimized is as follows:

$$J = E\left\{ \sum_{i=0}^N 1/2[\underline{x}^T(t_i)X\underline{x}(t_i) + \underline{u}^T(t_i)U\underline{u}(t_i)] + 1/2\underline{x}^T(t_{N+1})X_f\underline{x}(t_{N+1}) \right\} \quad (2-23)$$

where:

J = cost function to be minimized

X = state weight matrix

X_f = final state weight matrix

U = control weight matrix

N = number of sample periods from t_0 to t_N

t_{N+1} = final time

t_N = last time a control is applied and held constant
over the next sample period

The controller is steady-state constant-gain, with gains dependent upon the particular value of the parameter set used in the design. The LQ controller is developed fully in Appendix A.

Three estimator/controller combinations have been researched in past theses. First, the estimator provides only a state vector estimate to a fixed-gain controller which is designed around a nominal value of the uncertain parameter set, a_{nom} . The controller algorithm is of the form:

$$\underline{u}(t_i) = -G_C^* [\underline{a}_{nom}] \hat{\underline{x}}(t_i^+) \quad (2-24)$$

The second design method is for the estimator to provide parameter and state vector estimates to a controller with gains that are dependent on the parameter estimate:

$$\underline{u}(t_i) = -G_C^* [\hat{\underline{a}}(t_{i-1})] \hat{\underline{x}}(t_i^+) \quad (2-25)$$

where the parameter estimate generated at the previous sample time is used in order to reduce computational delay.

The third approach is to implement an elemental controller for each of the elemental filters of the sliding bank. The control outputs are probabilistically weighted, similar to Equation (2-12), to form:

$$\underline{u}(t_i) = \sum_{j=1}^J p_j(t_i) \underline{u}_j(t_i) \quad (2-26)$$

where:

$$\underline{u}_j(t_i) = -G_c^* [a_j] \hat{x}_j(t_i^+) \quad (2-27)$$

This is usually referred to as a multiple model adaptive controller (MMAC) [23:253]. This research uses the last approach. However, initially the controller design shown in Equation (2-27) was also going to be implemented in this research. The nominal controller was needed because if adaptive control inputs were supplied to the system before an accurate estimate of the true parameter was obtained, the control inputs could drive the system unstable. Therefore, a nonadaptive controller was designed that was very robust to parameter variation at the expense of controller performance so that the controller inputs

would at least maintain stability regardless of the estimate of the true parameter.

2.5. Ambiguity Function Analysis [3; 6; 9; 20:97-99]

Ambiguity function analysis can provide information about the performance of an estimator. The generalized ambiguity function is given by:

$$A_i(\underline{a}, \underline{a}_t) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} L[\underline{a}, \underline{z}_i] f_{\underline{z}(t_i) | \underline{a}(t_i)}(\underline{z}_i | \underline{a}_t) d\underline{z}_i \quad (2-28)$$

where \underline{a} is the parameter vector, \underline{a}_t is the true parameter vector, and $L[\underline{a}, \underline{z}_i]$ is a likelihood function upon which a parameter estimate would be based via maximum likelihood techniques. For a given value of \underline{a}_t , the curvature of the function of \underline{a} , at the value of \underline{a}_t , provides information on the ability of the filter to estimate that parameter: the sharper the curvature, the greater the precision. This curvature is inversely related to the Cramer-Rao lower bound on the estimate error covariance matrix by

$$E[(\underline{a} - \underline{a}_t)(\underline{a} - \underline{a}_t)^T] \geq [-(\partial^2 / \partial \underline{a}^2) A_i(\underline{a}, \underline{a}_t) |_{\underline{a}=\underline{a}_t}]^{-1} \quad (2-29)$$

The ambiguity function value $A_i(\underline{a}, \underline{a}_t)$ for any \underline{a} and \underline{a}_t can be calculated from the output of a conventional nonadaptive Kalman filter sensitivity analysis [22:97-99] in which the "truth model" is identical to the model upon

which the Kalman filter is based, except that they are based on \underline{a}_t and \underline{a} , respectively. The ambiguity function is then given by

$$\begin{aligned}
 A_i^M(\underline{a}, \underline{a}_t) = & \sum_{j=1-M+1}^m \left((-m/2 \ln(2\pi) - 1/2 \ln|A(t_j; \underline{a})|) \right. \\
 & - 1/2 \operatorname{tr} \left\{ A^{-1}(t_j; \underline{a}) [H(t_j) P_e(t_j^-; \underline{a}_t, \underline{a}) H^T(t_j) + R(t_j)] \right\} \\
 & - n/2 \ln(2\pi) - 1/2 \ln|P(t_i^+; \underline{a})| \\
 & \left. - 1/2 \operatorname{tr} [P^{-1}(t_i^+; \underline{a}) P_e(t_i^+; \underline{a}_t, \underline{a})] \right) \quad (2-30)
 \end{aligned}$$

where:

$A(t_i, \underline{a}) = [H(t_i) P(t_i^-; \underline{a}) H^T(t_i) + R(t_i)]^{-1}$
for the Kalman filter based on \underline{a} ,

$P_e(t_i^+; \underline{a}_t, \underline{a})$ = the covariance matrix of the error between the state estimate of the Kalman filter based on \underline{a} and the states of the true system based on \underline{a}_t , where " " or "+" denotes before or after incorporation of the i th measurement

m = the number of measurements

n = the number of states

M = the number of the most recent residuals included in the likelihood function (rather than using all residuals back to the first sample term)

The terms are summed over the most recent M sample times [22:98]; however, here M is set equal to one. This approximation can be made because the summation term is the component of Equation (2-30) that is sensitive to

parameter variation, while the rest of Equation (2-30) is sensitive to the state estimation. Therefore, the general shape of the curve can be found by using only the first term in the summation term (see Equation 2-31). This reduces the size of the fluctuations in the value of $A_1^M(\underline{a}, \underline{a}_t)$. Consequently, this flattens the surface of the plot of the ambiguity function plotted as a function of the parameter \underline{a} . The main benefit of setting $M = 1$ is that this significantly reduces the number of computations.

Filios encountered numerical difficulties while evaluating the ambiguity function. The numerical difficulties were caused by numerical precision problems encounter when cascaded multiplies were used in determinant evaluation [3:64]. Therefore, it is impossible to compute the ambiguity function as described in Equation (2-30). The numerical difficulties are overcome by approximating the expressions for the probability weighting factors (Equation 2-14) and the ambiguity function (Equation 2-30). Equation (2-14) is approximated as

$$f_j(\underline{z}(t_1)) = \exp[-(1/2)\underline{z}_j^T(t_1)A_j^{-1}(t_1)\underline{z}_j(t_1)] \quad (2-31)$$

This is no longer a true density function because the

scale factor is incorrect: however, because of the denominator in Equation (2-13), the probability weightings are still correct in the sense that they add to one [3:65]. If the determinants of the A matrices of the elemental filters are expected to be approximately equal in magnitude, in the absence of numerical problems, the relative magnitudes of the value of the ambiguity functions are not significantly altered [3:65]. Equation (2-30) is approximated by removing the terms containing the determinants of $P(t_i^+)$ and $A(t_i)$. Equation (2-30) becomes

$$\begin{aligned}
 A_i^1(a, a_t) = & -m/2 \ln(2\pi) - n/2 \ln(2\pi) \\
 & -1/2 \operatorname{tr}[A^{-1}(t_i; a)[H(t_i)P_e(t_i^-; a_t, a)H^T(t_i) + R(t_i)]] \\
 & -1/2 \operatorname{tr}[P^{-1}(t_i^+; a)P_e(t_i^+; a_t, a)] \quad (2-32)
 \end{aligned}$$

However, if M is equal to one, then the last term is ignored and Equation (2-32) becomes:

$$\begin{aligned}
 A_i^1(a, a_t) = & -m/2 \ln(2\pi) \\
 & -1/2 \operatorname{tr}[A^{-1}(t_i; a)[H(t_i)P_e(t_i^-; a_t, a)H^T(t_i) + R(t_i)]] \quad (2-33)
 \end{aligned}$$

This is a reasonable approximation since the determinants of $P(t_i^+)$ and $A(t_i)$ have minimal effect on the ambiguity function. Therefore, its primary sensitivity is in the

functions that are being preserved [3:66]. Karnick used this approximation in his thesis and the ambiguity plots he produced contained numerous peaks and valleys, making it very difficult to determine the correct space discretization.

However, there are other techniques that can be used to determine the discretization level. First, insight can be gained from the values of the dynamics noise strength (Q) and the measurement noise covariance (R). Large Q and R values indicate a need for large space discretization because such large Q and R values mask differences in the filters in a tightly spaced discretization. Another technique is to conduct a sensitivity study by giving the filter an incorrect parameter value and then observing the state estimate. The more inaccurate the state estimate becomes, for a given size variation between filter-assumed and true parameter values, the tighter the space discretization needs to be for that parameter.

2.6. Summary

This chapter develops algorithms necessary for implementation of the full-scale and moving-bank multiple model adaptive estimator and an appropriate adaptive controller based on this type of estimation. The moving bank MMAE is expected to yield significant computational savings over the full-scale MMAE, if they are both based

on the same discretization level. However, a viable alternative to a moving bank, which is also investigated, is a fixed-bank algorithm based upon a coarser discretization. The ambiguity function analysis is also developed.

Ambiguity functions are expected to give insight into the parameters that need adaptive estimation and into the appropriate levels of discretization of the parameter space to perform such estimation.

III. Rotating Two-Bay Truss Model

3.1. Introduction

This chapter develops the system equations for a large flexible space structure modeled as a two-bay truss. The truss is allowed to rotate around a fixed point, thereby incorporating both rigid body rotation and bending mode dynamics. The system model is first developed in physical variables and then transformed into the modal variable format. This is followed by a complete physical description of the two-bay truss. This model is taken from Karnick's thesis [9]. Section five motivates the need for order reduction and develops the order reduction technique used in this thesis.

3.2. Second Order and State Space Form Models

The forced vibration of a large space structure with active controls and n frequency modes can be described with the following general second-order differential equations [17; 32]:

$$M\ddot{\underline{r}}(t) + C\dot{\underline{r}}(t) + K\underline{r}(t) = \underline{F}_1(u,t) + \underline{F}_2(t) \quad (3-1)$$

where:

M = constant $n \times n$ mass matrix

C = constant $n \times n$ damping matrix

K = constant $n \times n$ stiffness matrix

$\underline{r}(t)$ = n -vector representing structure's physical coordinates

$\underline{E}_1(u, t)$ = control input

$\underline{E}_2(t)$ = disturbances

The control system is assumed to consist of a set of discrete actuators. The location of the actuators are given in Section 3.4.4. The external disturbances are represented by white noise, thus producing:

$$M\ddot{\underline{r}}(t) + C\dot{\underline{r}}(t) + K\underline{r}(t) = -b\underline{u}(t) - g\underline{w} \quad (3-2)$$

where " $\underline{\quad}$ " denotes a vector stochastic process and:

$\underline{u}(t)$ = vector of dimension m representing actuator input

b = $n \times m$ matrix identifying position and velocity relationship between actuators and controlled variables [17]

\underline{w} = vector of dimension r representing dynamics driving noise, where r is the number of noise inputs

g = $n \times r$ matrix identifying position and relationships between dynamics driving noise and controlled variables.

The state representation of Equation (3-2) can be written as:

$$\dot{\underline{\tilde{x}}} = F\underline{\tilde{x}} + B\underline{u} + G\underline{w} \quad (3-3)$$

where the $2n$ -dimensional vectors $\underline{\tilde{x}}$ and $\dot{\underline{\tilde{x}}}$ are given by:

$$\underline{\tilde{x}} = \begin{bmatrix} \underline{\tilde{r}} \\ \underline{\dot{\tilde{r}}} \end{bmatrix} \quad \dot{\underline{\tilde{x}}} = \begin{bmatrix} \underline{\dot{\tilde{r}}} \\ \underline{\ddot{\tilde{r}}} \end{bmatrix} \quad (3-4)$$

and the open-loop plant matrix F , the control matrix B , and the noise matrix G are given by:

$$F = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}_{2n \times 2n} \quad B = \begin{bmatrix} 0 \\ -M^{-1}b \end{bmatrix}_{2n \times m} \quad (3-5)$$

$$G = \begin{bmatrix} 0 \\ -M^{-1}g \end{bmatrix}_{2n \times r}$$

It is assumed that the noise enters the system at the same place as the actuators; therefore, the b matrix is equal to the g matrix and m is equal to r . In addition, it is assumed measurements are available from accelerometers and gyros. The location of the sensors are given in Section 3.4.4. The output signals of the accelerometers are integrated once to provide velocity measurements and once again to provide position measurements. Therefore, the position and velocity measurements are assumed co-located.

These position and velocity measurements are then fed into the system model. It is also assumed that the roundoff errors due to the A/D conversion associated with the two integrations are small and can be accounted for by increasing the measurement noise (R). In addition, it is assumed that the position and velocity measurements are noise corrupted due either to deficiencies in the model of the sensor or some actual external measurement noise. The position and velocity measurements are modeled as:

$$\underline{z} = \begin{bmatrix} H & 0 \\ 0 & H' \end{bmatrix} \underline{x} + \underline{v} \quad (3-6)$$

px2n

where:

p = the number of measurements

\underline{v} = an uncertain measurement disturbance of dimension p and modeled as a white Gaussian noise [21:114]

H = position measurement matrix

H' = velocity measurement matrix

Since the position and velocity measurements are co-located, the measurement matrices are equal ($H = H'$).

3.3. Modal Analysis

Modal analysis is used to transform the system into a set of independent equations by transforming the system from physical coordinates to modal coordinates. In order to achieve decoupling, the damping matrix must be assumed to be a linear combination of the mass and stiffness matrices [16]:

$$C = \alpha K + \beta M \quad (3-7)$$

The determination of α and β is discussed later in this section.

The modal coordinates are related to the physical coordinates by

$$\underline{r} = T \underline{\Omega} \quad (3-8)$$

where \underline{r} is as defined previously and $\underline{\Omega}$ represents the modal coordinates. The matrix, T , is an $n \times n$ matrix of eigenvectors and is the solution to [17; 27; 32; 33]:

$$\omega^2_{MT} = KT \quad (3-9)$$

The values of ω which satisfy Equation (3-9) are natural or modal frequencies [8:66]. Substituting Equation (3-8) into Equation (3-3) gives

$$\dot{\underline{x}}' = F' \underline{x}' + B' \underline{u} + G' \underline{w} \quad (3-10)$$

where \underline{x}' is now defined as:

$$\underline{x}' = \begin{bmatrix} \underline{c} \\ \underline{c} \end{bmatrix}_{2n \times 1} \quad \dot{\underline{x}}' = \begin{bmatrix} \dot{\underline{c}} \\ \dot{\underline{c}} \end{bmatrix}_{2n \times 1} \quad (3-11)$$

and the open loop plant matrix F' , the control matrix B' , and the noise matrix G' are:

$$F' = \begin{bmatrix} 0 & I \\ -T^{-1}M^{-1}KT & -T^{-1}M^{-1}CT \end{bmatrix}_{2n \times 2n}$$

$$B' = \begin{bmatrix} 0 \\ -T^{-1}M^{-1}b \end{bmatrix}_{2n \times m} \quad G' = \begin{bmatrix} 0 \\ -T^{-1}M^{-1}b \end{bmatrix}_{2n \times m} \quad (3-12)$$

The F' matrix can also be written in the following form [32; 33]:

$$F' = \begin{bmatrix} 0 & I \\ [-\omega_i^2] & [-2\zeta_i \omega_i] \end{bmatrix}_{2n \times 2n} \quad (3-13)$$

where each of the four partitions are nxn dimensional and diagonal and where ω_i and ζ_i are the undamped natural frequency and the damping ratio of the ith mode. The measurements become:

$$\underline{\dot{z}} = \begin{bmatrix} HT & 0 \\ 0 & HT \end{bmatrix} \underline{z} + \underline{v} \quad (3-17)$$

Since the system is formulated in modal coordinates [17], it is assumed that uniform damping exists throughout the structure. The level of structural damping is determined by selecting a value for the damping coefficients (ζ_i) and substituting this value into Equation (3-13). The particular value of the damping coefficient has no effect on the calculation of ω_i since it is the natural or modal frequency. The assumption simplifies the determination of structural damping and allows a better physical insight into formulating the problem than does the selection of values for α and β as shown in Equation (3-7). The damping coefficient of $\zeta_i = 0.005$ is chosen for implementation because it is characteristic of damping associated with large space structures [17; 24].

3.4. Two-Bay Truss

3.4.1. Introduction. This section describes the physical structure of the two-bay truss rotating about a

fixed point. This section discusses the physical dimensions of the model, the analysis used to develop the two-bay truss model, the sensors and actuators, and the physical parameter variations to be considered in the model.

3.4.2 Background. For this research a fixed two-bay truss, which was used to research active control laws for vibration damping [17] (see Figure 3-1), was modified twice to produce the system model. First, non-structural masses were added to nodes 1 through 4, which reduced the structural frequencies, thereby making the problem more like a large space structure [17]. Second, rigid body motion was added to the truss (see Figure 3-2). These modifications produced a model which approximated a space structure that has a hub with appendages extending from the structure. The mass of the hub was large relative to the mass of the appendage and the hub can then be rotated to point the appendage in a commanded direction.

The rigid body motion was established by adding a point (node 7) that remained fixed while the two-bay truss was free to rotate about this point in the x-y plane; see Figure 3-2. The truss was connected to this point with rods having radii that were large relative to the rods used to construct the two-bay truss. Making the rods large made a very "stiff" link between the truss and node 7, which

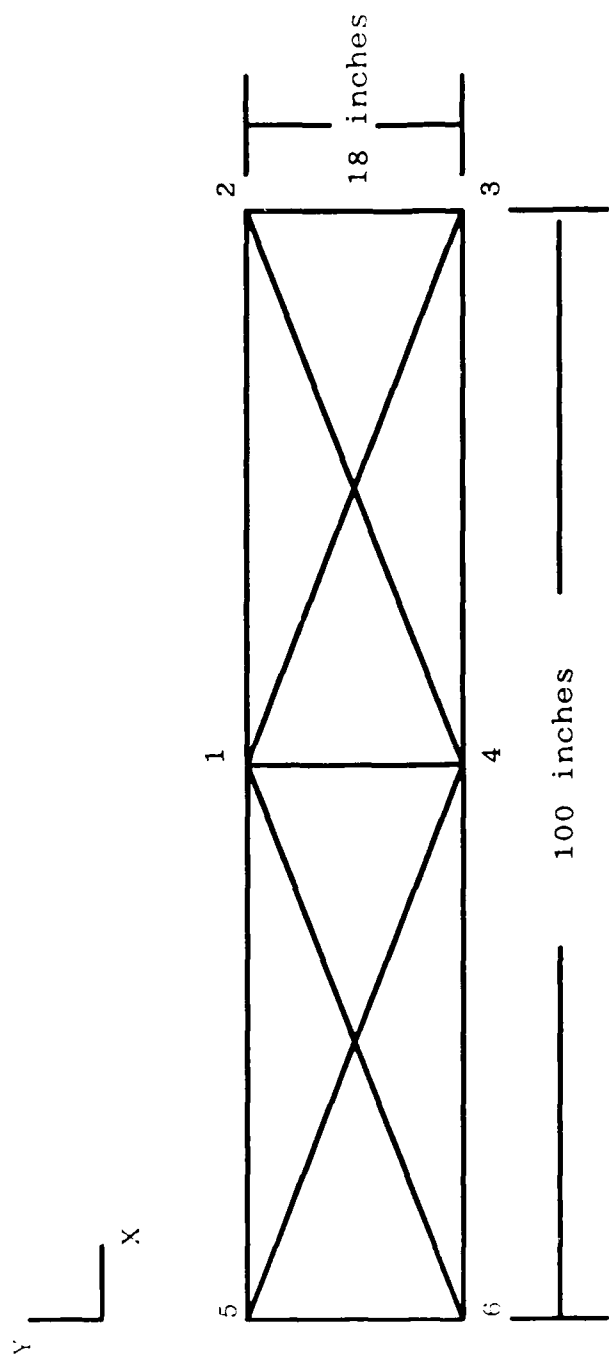


Fig. 3-1. Two-bay Truss Model

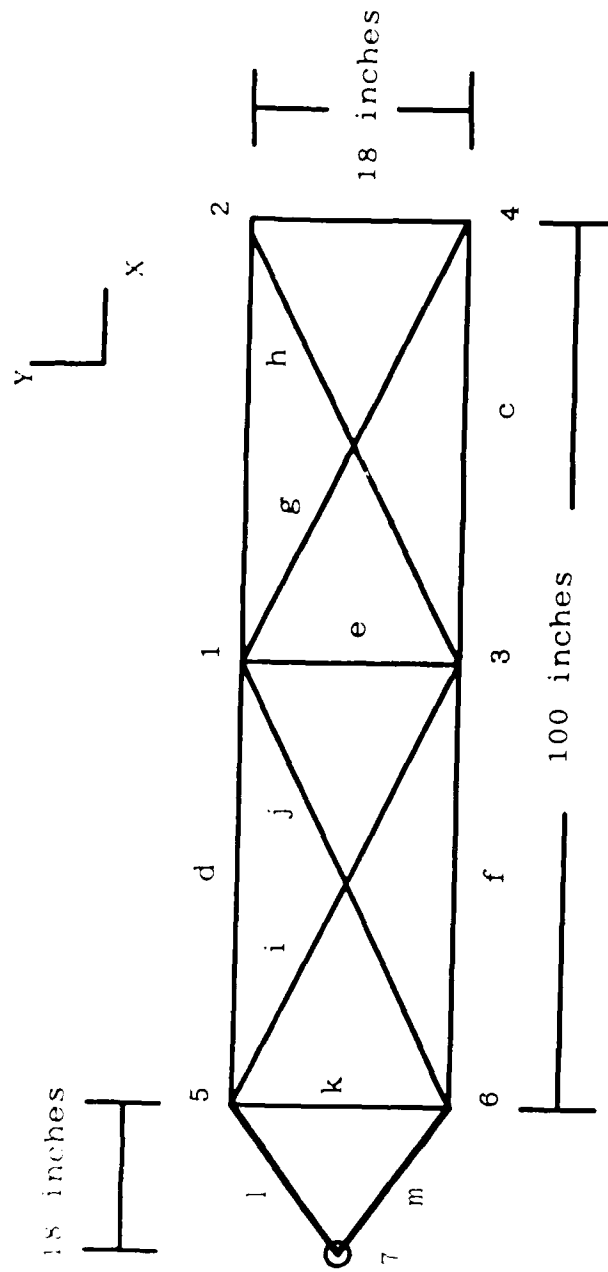


Fig. 3-2. Rotating Two-bay Truss Model

introduced high frequency modes into the structure but kept the lower modal frequencies similar to the case where the truss was fixed.

3.4.3. Two-Bay Truss Construction. The structure consisted of 13 rods which were assumed to be constructed of aluminum, having a modulus of elasticity of 10 psi and weight density of 0.1 lb/in [32]. The cross-sectional areas of each member shown in Figure 3-2 are given in Table 3-1.

TABLE 3-1

STRUCTURAL MEMBER'S CROSS-SECTIONAL AREAS

Member	Area (in ²)	Member	Area (in ²)
a	.00321	h	.00328
b	.00100	i	.00439
c	.00321	j	.00439
d	.01049	k	.20000
e	.00100	l	.20000
f	.01049	m	.20000
g	.00328	-	-

The cross-sectional areas of rods a through j were calculated by optimizing the weight of the structure shown in Figure 3-1. The large space structure was calculated with its weight minimized subject to the constraint that the fundamental frequency remain unchanged. This constraint was needed so that the natural frequency response of the system remained constant. Rods k through m were used to

make the "stiff" link between node 7 and the two-bay truss. In order to achieve this stiffness, the areas of rods k and l were arbitrarily selected to be large relative to the area of other rods.

Non-structural masses of $1.294 \text{ lb-sec}^2/\text{in}$, were located at positions 1, 2, 3, and 4 as shown in Figure 3-2. The non-structural mass was very large compared to the structural mass to achieve the low frequencies associated with large space structures [17]. The actual value of the non-structural mass was selected using an optimization technique [33], which found the mass necessary to attain a frequency of 0.5 Hz in the lowest mode for the fixed two-bay truss [17].

The mass and stiffness matrices, describing the system model, were obtained using finite element analysis [31], which models a structure as consisting of a finite number of nodes connected by elements. The software program that was used [33] has the capability to use a number of different elements. In this research effort, the rods were described by cross-sectional area, modulus of elasticity, and weight density. The finite element program produced mass and stiffness matrices with dimension equal to the number of degrees of freedom (DOF) associated with the model. Each row of the mass and stiffness matrix was associated with a specific node and DOF. For the two-bay

truss shown in Figure 3-2, row 1 of each mass and stiffness matrix is associated with the x-axis DOF of node 1. Each node actually has three translational DOF. However, only planar motion was being considered (in order to simplify the model); therefore, the nodes were modeled with only two DOF. For this problem, node 7 was fixed. Therefore, all three DOF associated with this node were eliminated, thereby reducing the dimensionality of the mass and stiffness matrices to 12 states yielding a 24-state model. The mass and stiffness matrices for the specifications previously discussed, are listed in Appendix A. These were the nominal matrices from which parameter variations were considered.

3.4.4. Sensors and Actuators. Accelerometers were located at nodes 1 and 2 as shown in Figure 3-2. The accelerometers should not be located at a node of the bending modes being detected because the displacement of the truss caused by the bending modes can not be detected. The accelerometer outputs were integrated twice to produce velocity and position measurements. There were also two additional gyro sensors co-located on the hub (node 7) of the two-bay truss. These sensors provided angular displacement and velocity. Actuators were placed at nodes 1 and 2 as shown in Figure 3-2. The actuators do not have to be co-located with the sensors, however they are in this

case to simplify the system model. These actuators were assumed to be force sensors (thrusters, etc.). An additional actuator was located on the hub and is assumed to be an inertia wheel.

The states corresponding to velocity and position were directly available in physical variable formulation, (Equations 3-3, 3-4, 3-5) while the states corresponding to angular displacement and velocity were directly available in modal formulation (Equations 3-10, 3-11, 3-12, 3-14). The H and b matrices were constructed by calculating separate matrices in the different state space formulations. These matrices were augmented after the physical variable formulations have been transformed into modal coordinates.

3.4.5. Physical System Parameter Uncertainty. The purpose of this thesis was to test the moving-bank multiple model adaptive estimation and control algorithms. Therefore, the model must have parameter uncertainty which allowed adaptive estimation to be applied. A 10 by 10 point parameter space was created by considering two physically motivated parameter variations. The parameters that were varied were the masses of the four non-structural masses and the stiffness matrix. The weight variation can be physically related to fuel being expended from or added to a tank or weight being shifted to a different section

(other than the two-bay truss) of the space structure. The stiffness variation can be associated with structural fatigue in the rods or a failure of a member within the structure itself. The exact parameter discretization will be determined in a sensitivity study performed in this thesis. Initially in forming the parameter space, the four non-structural mass were multiplied by constant terms that ranged from 0.5 to 1.4, while the stiffness matrix was multiplied by constant terms that ranged from 0.8 to 1.16. However following the space discretization study, the masses were multiplied by constant terms that ranged from 0.5 to 1.5, while the stiffness matrix was multiplied by constant terms that ranged from 0.5 to 1.4. The mass and stiffness matrices were chosen as parameters because they appeared in the system equations and made logical sense; however, with further study more appropriate parameters may be determined. An additional technique used in determining the appropriate uncertain parameters is ambiguity function plots [3].

3.5. State Reduction

3.5.1. Introduction. The mass and stiffness matrices were previously shown to be of dimension 12. This produces a system model that has 24 states, which was much larger than desired for the control synthesis of this thesis effort and for a practical control application.

This section develops a method of order reduction referred to as singular perturbations [10; 11; 17; 23:219], which assumes that faster modes reach steady state essentially instantaneously. This section describes the method of singular perturbations and then discusses the magnitude of the order reduction.

3.5.2 Development. The deterministic system is reformulated as follows:

$$\begin{bmatrix} \dot{\underline{x}}_1 \\ \dot{\underline{x}}_2 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \underline{u} \quad (3-15)$$

$$\underline{z} = [H_1 \ H_2] \underline{x} \quad (3-16)$$

The \underline{x}_1 states are to be retained and F_{11} and F_{22} are square matrices. If the high frequency modes, represented by \underline{x}_2 states, are eliminated, steady state is assumed to be reached instantaneously for these modes ($\dot{\underline{x}}_2 = \underline{0}$). The \underline{x}_2 states are then expressed in terms of the \underline{x}_1 states and controls, where it is assumed F^{-1} exists, as follows:

$$\dot{\underline{x}}_2 = \underline{0} = F_{21}\underline{x}_1 + F_{22}\underline{x}_2 + B_2\underline{u} \quad (3-17)$$

$$\underline{x}_2 = -F_{22}^{-1}(F_{21}\underline{x}_1 + B_2\underline{u}) \quad (3-18)$$

Substituting for \underline{x}_2 gives

$$\dot{\underline{x}}_1 = \underline{F}_r \underline{x}_1 + \underline{B}_r \underline{u} \quad \underline{z} = \underline{H}_r \underline{x}_1 + \underline{D}_r \underline{u} \quad (3-19)$$

where:

$$\underline{F}_r = (\underline{F}_{11} - \underline{F}_{12} \underline{F}_{22}^{-1} \underline{F}_{21}) \quad (3-20)$$

$$\underline{B}_r = (\underline{B}_1 - \underline{F}_{12} \underline{F}_{22}^{-1} \underline{B}_2) \quad (3-21)$$

$$\underline{H}_r = (\underline{H}_1 - \underline{H}_2 \underline{F}_{22}^{-1} \underline{F}_{21}) \quad (3-22)$$

$$\underline{D}_r = (-\underline{H}_2 \underline{F}_{22}^{-1} \underline{B}_2) \quad (3-23)$$

Note that the \underline{D}_r matrix did not exist before order reduction. It is a direct-feed forward term which is not in the unreduced system [17].

This order reduction technique is now applied to a system of the form of Equation (3-10). Reordering Equation (3-13) into the reduced-order form produces Equation (3-24), where the upper partition contains the modes to be retained while the lower partition contains those assumed to reach steady state instantaneously:

$$F = \left[\begin{array}{cc|cc} 0 & I & & \\ [-\omega_1^2] & [-2\zeta_1\omega_1] & & 0 \\ \hline & & 0 & I \\ 0 & & [-\omega_2^2] & [-2\zeta_2\omega_2] \end{array} \right]_{2n \times 2n} \quad (3-24)$$

The exact dimension of ω_1 is equal to two times the number of modes contained in the system model. The exact dimension of ω_2 is equal to two times the number of modes assumed to reach steady state. Comparing Equation (3-24) to Equation (3-15) shows that the partitions F_{12} and F_{21} are zero. Substituting this result into Equation (3-19) yields:

$$F_r = F_{11} \quad (3-25)$$

$$B_r = B_1 \quad (3-26)$$

$$H_r = H_1 \quad (3-27)$$

$$D_r = (-H_2 F_{22}^{-1} B_2) \quad (3-28)$$

D_r is the only term in Equation (3-19) to Equation (3-24) that is dependent upon terms associated with the states assumed to reach steady state instantaneously. The other reduced-order matrices are calculated simply by truncating those states associated with \underline{x}_2 .

Calculation of D_r can be greatly simplified by examining the form of Equation (3-28). H_2 is similar in form to Equation (3-6):

$$H_2 = \begin{bmatrix} H_C & 0 \\ 0 & H'_C \end{bmatrix} \quad (3-29)$$

H_C represents measurement of the unmodeled position states while H'_C represents measurement of the unmodeled velocity states. In Equation (3-6), it is assumed that the position and velocity measurement matrices are identical because the measurements are derived from an acceleration measurement. The same assumption can be made in Equation (3-29); however, the distinction between the velocity and measurement matrices is retained since it is shown in Equation (3-33) to be important in the general development of the reduced order matrices. As is shown in Equation (3-27), F_{22} is a square matrix of the form:

$$F_{22} = \begin{bmatrix} 0 & I \\ [-\omega_2^2] & [-2\zeta_2\omega_2] \end{bmatrix} \quad (3-30)$$

where each of the four partitions is a square, diagonal matrix whose dimension is dependent upon the number of states to be retained. Its inverse is [8]:

$$F_{22}^{-1} = \begin{bmatrix} [-\omega_2^2]^{-1} [2 \omega_2 f_2] & [-\omega_2^2]^{-1} \\ I & 0 \end{bmatrix} \quad (3-31)$$

B_2 is similar in form to the matrix B described in Equation (3-5):

$$B_2 = \begin{bmatrix} 0 \\ b' \end{bmatrix} \quad (3-32)$$

where b' represents the rows of the matrix product $-M^{-1}b$ corresponding to the unmodeled states. Evaluation of Equation (3-28) yields:

$$D_r = \begin{bmatrix} H_c [-\omega_2^2]^{-1} b' \\ 0 \end{bmatrix}_{p \times m} \quad (3-33)$$

where p is the number of measurements. Only the position measurements are affected since the lower portion is zero. The D_r matrix is only dependent upon the position portion of the measurement matrix and not the velocity measurement matrix. The inverse of $-\omega_2^2$ is easily calculated since the matrix is diagonal. An example of detailed system matrix development and order reduction is listed in Appendix B.

3.5.3. Order Reduction Selection. The number of modes retained was determined by examination of the eigenvalues and frequencies of the unreduced system (Table 3-2). The eigenvalues were determined for an undamped system. This greatly simplified the problem and was a good approximation because large space structures have very small damping coefficients ($\zeta_i = 0.005$). The frequencies can be distinctly divided into several groups of closely spaced frequencies. For example, modes 4, 5, and 6 were clearly one set of closely spaced frequencies. When reducing the order of system by the method of singular perturbations, it is not desired to make the reduction at a point which divides a group of "closely spaced" frequencies [24]. At the same time, a sufficient number of frequencies must be retained in order to do an adequate job of estimation and control. In this research, the truth models for simulating the real world and design models upon which to base the estimator/controller algorithms were investigated at dimensions of 6 for the truth models and 6 for the filter models. Therefore, this research did not analyze the effect of order mismatch between the filter models and the truth model.

3.6. Summary

This chapter developed the system equations for the two-bay truss with rigid body motion. The mathematical

TABLE 3-2
EIGENVALUES AND FREQUENCIES

Mode No.	Eigenvalues	Frequencies (Hz)
1	0.0000	0.0000
2	8.8922	1.4152
3	22.5492	3.5888
4	29.5444	4.7021
5	31.1519	4.9580
6	32.8002	5.2203
7	54.3893	8.6563
8	58.1592	9.2563
9	985.9204	156.9141
10	9018.8987	1435.4023
11	11515.9941	1832.8274
12	19956.5072	3176.1768

The eigenvalues are for an undamped system ($\zeta_i = 0$).

model was dependent upon physical parameters which, in reality, vary from those used in the mathematical model. The moving-bank MMAE was used to estimate both the reduced order system states and varying parameters of the physical system.

IV. Simulation Plan [3; 6; 9]

4.1. Introduction

Evaluation of the performance of the moving-bank multiple model adaptive estimator/controller for this application requires simulating actual space structure movement and estimator/controller operation. The computer simulations provide both a Monte Carlo analysis and a sensitivity analysis (using ambiguity functions) of the estimator/controller. This chapter provides background on the Monte Carlo simulation, briefly outlines the computer software, and then discusses the simulation plan for analyzing the performance of the estimator design logics and the moving-bank algorithms.

4.2. Monte Carlo Analysis

The objective of this analysis is to obtain statistical information on the estimator/controller's performance, through the use of a Monte Carlo study. A Monte Carlo simulation must be used instead of a covariance simulation, even though the system is linear, because of the adaptive nature of the problem. The Monte Carlo study obtaining many samples of the error process through simulation and then using this data to approximate the process statistics [21:329].

The true system model under consideration can be described by a linear time-invariant difference equation:

$$\underline{\tilde{x}}(t_{i+1}) = \Phi(t_{i+1}, t_i) \underline{\tilde{x}}(t_i) + B_d(t_i) \underline{u}(t_i) + G_d(t_i) \underline{w}_d(t_i) \quad (4-1)$$

(See Equation (2-1) for a complete definition of terms). B_d and G_d are the discrete-time equivalents [22:171] of the B and G matrices given in Equation (3-5). It is assumed that the noise input matrix is identical to the control input matrix; therefore [21:171],

$$G_d = I$$

$$B_d = \int_{t_{i-1}}^{t_i} \Phi(t_i, \tau) B \, d\tau \quad (4-2)$$

Noise-corrupted measurements are provided to the estimator in the form of:

$$\underline{\tilde{z}}(t_i) = H \underline{x}(t_i) + \underline{v}(t_i) \quad (4-3)$$

where H is the measurement matrix and $\underline{v}(t_i)$ is a discrete time, zero-mean, white Gaussian measurement noise with covariance matrix R . Matrices Φ , B_d , G_d , and H are functions of the true parameter vector \underline{a}_t :

$$\underline{a}_t = [M_t, K_t]^T$$

where M_t and K_t are the true mass and stiffness parameters, respectively and are discussed in Section 3.4.

The simulation is accomplished for a sufficient number of runs so that the computed sample means and variances of the random variables of interest are good approximations to true ensemble averages. The number of simulation runs selected is 10, and this is determined as sufficient by observation of the sample statistics as the number of Monte Carlo runs is increased [3:52].

Figure 4-1 illustrates the block diagram for the Monte Carlo simulation [3:53; 22]. The variables not previously defined are:

$\underline{x}_t(t_i)$ = "truth model" states

$\hat{\underline{x}}(t_i)$ = estimate of system states

$\hat{\underline{a}}(t_i)$ = estimate of uncertain parameter vector

$\underline{e}_a(t_i)$ = error in the parameter estimate

$$\underline{e}_a(t_i) = \underline{a}_t(t_i) - \hat{\underline{a}}(t_i)$$

$\underline{e}_x(t_i)$ = error in the system state estimate

$$\underline{e}_x(t_i) = T \underline{x}_t(t_i) - \hat{\underline{x}}(t_i)$$

where T is a $n \times n_t$ matrix to make the dimensions compatible, since the estimate is typically of lesser dimension than that of the "truth model" states.

The sample mean of the vector of variables of interest is computed as [3:53; 6:46; 21:129; 9:61]:

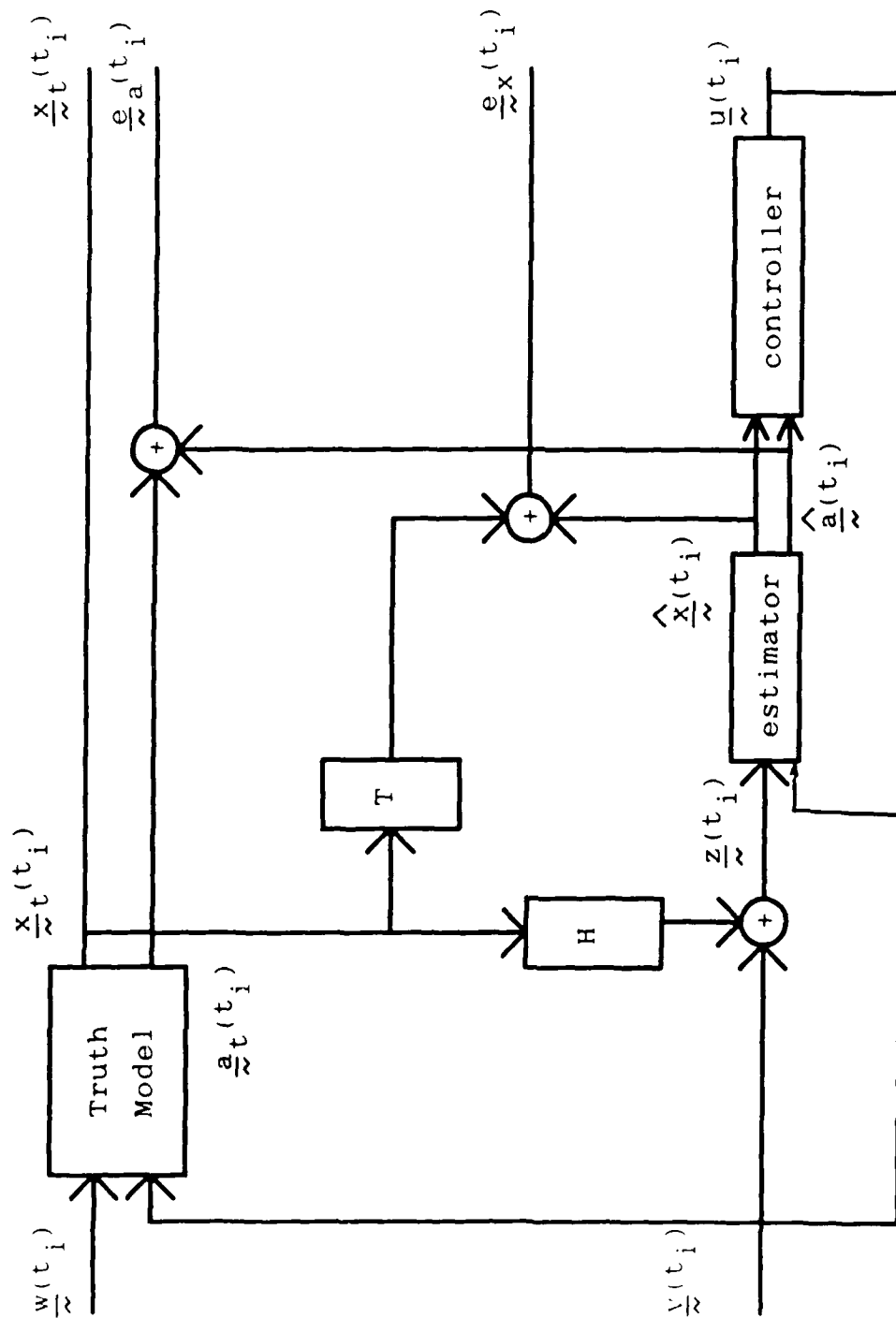


Fig. 4-1. System Estimator and Controller Simulation

$$E[\underline{e}_x(t_i)] \approx \underline{M}_{ex}(t_i) \triangleq (1/L) \sum_{k=1}^L \underline{e}_{xk}(t_i) \quad (4-4)$$

where:

L = total number of simulation runs

$\underline{e}_{xk}(t_i)$ = value of $\underline{e}_x(t_i)$ during the k th simulation run

The sample covariance of $\underline{e}_x(t_i)$ is computed as [3:54; 6:47; 21:130; 9:63]:

$$E\{[\underline{e}_x(t_i) - E\{\underline{e}_x(t_i)\}][\underline{e}_x(t_i) - E\{\underline{e}_x(t_i)\}]^T\} \triangleq P_{ex}(t_i) \approx$$

$$[1/(L-1)] \sum_{k=1}^L \underline{e}_{xk}(t_i) \underline{e}_{xk}^T(t_i) - [L/(L-1)] \hat{\underline{M}}_{ex}(t_i) \hat{\underline{M}}_{ex}^T(t_i) \quad (4-5)$$

Similar relationships for $\underline{e}_a(t_i)$ can also be generated readily.

When evaluating the estimator alone, the feedback controller in Figure 4-1 is replaced by a dither signal with a frequency and amplitude that is determined by trial and error. It has been shown that a dither signal can be used to excite the system model and enhance parameter identification [6:50, 58-59; 20:135, 136; 21:229; 9]. The error in the state estimate and the error in the parameter estimate are useful in evaluating the performance of the estimator. The error in the state estimate gives

the best means of comparing the estimator to other types of estimators, since the primary concern is accurate state estimation (and control eventually) rather than parameter estimation for its own sake. The error in the parameter estimate lends insight into the accuracy of state estimation, indicates the precision of a parameter estimate that may be fed to an adaptive controller, and provides a means of evaluating various move, contract, and expand algorithms [3:54].

When evaluating the estimator/controller combination, it is more appropriate to examine the statistics of the true state values. In this thesis effort, the control objective is to quell any oscillations in the two-bay truss structure and to "point" the two-bay truss in a commanded direction. It is also important to examine the magnitude of the control inputs in order to detect unreasonable commanded control levels. Lynch [16] in his research with a fixed two-bay truss (see Section 3.4.2) limited the magnitude of the force of an individual actuator to 100 lbs. Although this limit is not implemented in this thesis, it is used as a general guideline as to a reasonable range of actuator activity.

4.3. Software Description [3:55-59]

4.3.1. Introduction. The analysis of the moving-bank estimator/controller requires the development of

four computer programs. Each of these programs is a modification of programs developed by Hentz, Filios, and Karnick [3; 6; 9]. For a more detailed description of the following programs, the reader is referred to Filios [3]. The first program is a preprocessor which creates a parameter space that is utilized in the Monte Carlo simulation and ambiguity functions analysis (see Section 2.5). The second program simulates the moving-bank multiple model adaptive estimator and performs Monte Carlo simulation runs and generates data for each run. This program is also modified to produce Monte Carlo runs of individual elemental filters. This allows for sensitivity studies that are conducted on individual controllers, such as tuning the state weighting matrix (X) and control weighting matrix (U) for individual controllers. The third program is a post-processor that computes the means and variances of variables of interest and then generates the plots of statistics for the Monte Carlo simulation. The fourth program computes the ambiguity functions and generates their plots.

4.3.2. Preprocessor. The preprocessor computes the discretized, reduced order (equal to the truth model), system matrices (Φ , B_d , G_d , H , D_r in Equations (2-1) and (3-19)) for each parameter point within the parameter space, the Kalman filter and LQ controller gains, and information needed for the ambiguity functions analysis

(see Section 2.5). An input file allows the state and control weighting matrices (X and U), dynamic driving noise strength (Q), measurement noise covariance (R), as well as the time increment for the discrete system, to be varied. This input file also contains the mass and stiffness matrices and two vectors which specify the mass and stiffness variation. The mass and stiffness matrices are used to determine a 24-state system (see Equation 3-4) which is then reduced to the number of states specified (see Section 3.5).

4.3.3. Primary Processor. The primary processor performs the Monte Carlo simulations. For each sample period, the true system and the filters currently implemented in the moving bank are propagated forward from the most recent sample time. A noise corrupted measurement is then made of the true system, and the filters of the moving bank are updated. The program then calculates the necessary control inputs and makes decisions on whether to move, expand, or contract the bank. After each sample period is complete, the values of the variables of interest are written to a data file.

The inputs to the primary processor describe the parameter space (obtained from the preprocessor) and true system parameters, and specify the move/contract/expand algorithms to be implemented, the associated thresholds,

initial probability weightings for the filters in the moving bank, and initial filter states. The output of the primary processor is a data file for each variable of interest (state estimate, actual state values, control inputs) covering all of the simulation runs, and a more detailed print file covering just the first Monte Carlo run. The print file lists the exact filters implemented in the moving bank and the variables which affect the decision algorithms.

4.3.4. Postprocessor. The postprocessor takes the variable data files obtained from the primary processor and calculates the sample means and variances from t_0 to t_{N+1} . Plots are then generated which contain the time histories of the means and the standard deviation of each variable. The postprocessor is run for each data file generated by the primary processor.

4.3.5. Ambiguity Functions Analysis. The ambiguity function analysis involves two programs. The first evaluates the ambiguity function for each point in the parameter space (See Chapter II for a full discussion of ambiguity functions). A square wave dither signal is used to enhance the difference between filter models. The form, magnitude, and frequency of the dither was determined by trial and error. The input file to this program specifies the truth model, number of Monte Carlo runs, as well as the

time length of the simulation. The system matrices are obtained from the same file generated by preprocessor for the primary processor. The second program reads the information from a file and then generates a three-dimensional plot of the ambiguity function values versus location in the parameter space.

4.4. Simulation Plan

4.4.1. Noise Level Determination. The goal of the noise level determination study was to determine an appropriate value for the noise covariance (R) and dynamics noise strength (Q). Therefore, the Q and R values for the "truth" model and filter model were set equal and were assumed to be diagonal for simplicity. The parameter used for both the "truth" and filter model was (5,5), which corresponds to the center of the bank. This particular parameter was chosen because it is the center of the overall parameter space. For this study, the single filter primary processor consisted of a single filter matched against a "truth" model.

The Q values were determined by holding the R values constant and varying the Q values one at a time. The Q values were decreased by orders of magnitude until the rms error for the state estimate no longer decreased substantially. After determining the Q values, the R values were

found using the same technique but holding the Q values constant and varying the R values one at a time. Note that, during this study, the Q and R values for the "truth" and filter models always remain equal.

4.4.2. Two-State Model. Hentz showed that for a two-state (i.e. one bending mode) filter model matched against a two-state truth model, the center of the moving bank would move in the direction of the true parameter. The goal of the two-state model study was to reproduce Hentz's work [6] for the truss model. To accomplish this, the preprocessor was modified so that it produced a two-state (i.e., one bending mode) filter for each parameter point. The primary processor in the two-state model study used a parameter space that consisted of all 100 points upon which to base elemental Kalman filters. In addition the primary processor was modified so that a record could be kept of the movement of the estimated true parameter (a) (i.e. movement of the bank), throughout the parameter space. The record was acquired by initializing a 10x10 matrix of elements to zero and then adding unity at each sample time to the matrix element corresponding to the current estimate of the true parameter. Entries were recorded from $t = 1.0$ to the end of the run. Recording was not started at $t = 0.0$ in an attempt to avoid transient effects. In addition, the moving-bank used the smallest

discretization level and was initially centered at the (5,5) parameter point (i.e., the middle of the parameter space). Finally, the postprocessor was not used in this study because plots of the state variables were not required.

4.4.3. Six-State Filter. The next step in this research effort was to determine if system complexity affects the performance of the moving-bank algorithm. The preprocessor was returned to its original form of producing six-state filters. The filters consisted of two states per mode for one rigid body mode and two bending body modes. The primary processor required no modifications from the two-state study. Also, the postprocessor was not needed for this study because the plots of the state variables were not required.

4.4.4. Space Discretization. Following the six-state filter study, a space discretization study was conducted to determine an appropriate space discretization level. The preprocessor again produced six-state filters. However, as in the noise level determination study, the primary processor used a single filter in the bank matched against a "truth" model. Also, the postprocessor was modified to produce rms error plots. The space discretization study was conducted using the following steps:

1. Choose one parameter to vary and hold the other parameter constant.

2. Set the filter parameter equal to the middle value in the range of possible values for the parameter that is constant and to the smallest value in the range of possible values for the varying parameter.

3. Set the truth model parameter equal to the filter parameter value.

4. Obtain RMS error plots for this setup.

5. Set the truth model parameter equal to the next larger parameter value for the varying parameter.

6. Obtain rms error plots for this setup.

7. Compare the rms error plots from step 6 and step 4. If the rms error plots for step 6 are 20% to 30% larger than for step 4, leave the parameter point that corresponds to the filter in the parameter space and move the filter to the parameter point associated with the truth model (repeat steps 2 through 7). If the rms error plots for step 6 are less than 20% larger than the plots in step 4, remove the

parameter point associated with the truth model from the parameter space and move the truth model to the next larger parameter in parameter space (repeat steps 5 through 7). If the rms error plots for step 6 are greater than 30% larger than the plots in step 4, move the truth model halfway between the filter parameter and the truth model parameter (repeat steps 5 through 7).

8. Interchange the varying parameter and the constant parameter after the varying parameter space is determined. Then repeat the process for the other parameter.

Note: If the edge of parameter space is reached before 10 values for the parameter are determined, then augment the parameter space with larger parameter values.

The objective of this study was to obtain an rms error growth of 20% to 30% between each step in parameter space. The results of this study are contained in Section 5.5, with the eventual ranges of the mass and stiffness parameters shown in Figure 5-11.

4.4.5. Measurement and Dynamics Noise Sensitivity. The goal of this study was to determine the upper limits, whose existence is discussed in Section 1.4, on the Q and R values. Once again, the preprocessor

AD-A190 761

MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED
TO FLEXIBLE SPACES. (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R W LASHLEE

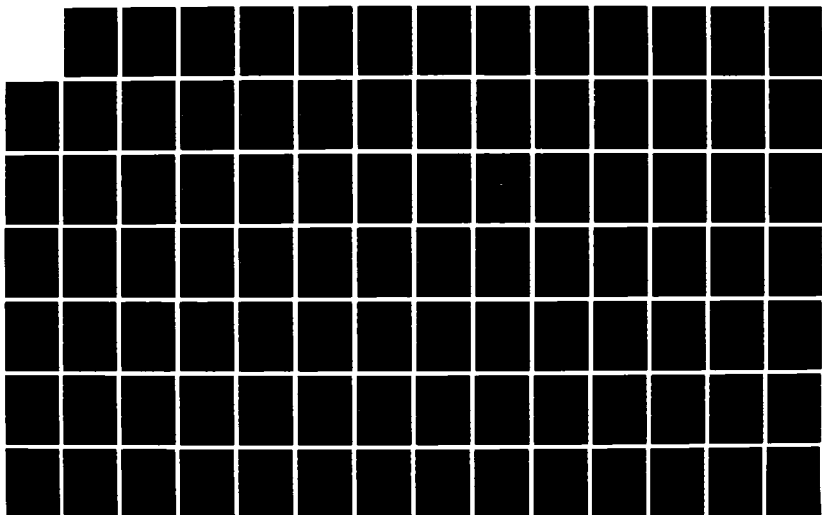
2/3

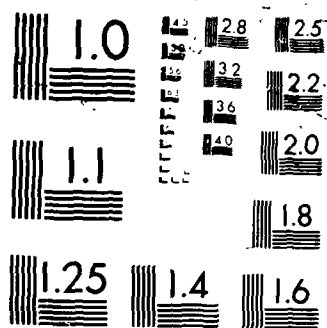
UNCLASSIFIED

DEC 87 AFIT/GE/ENG/87D-36

F/G 22/4

ML





produced six-state filters and the primary processor used a single filter matched against a truth model. As in the noise level determination, the measurement noise covariances (R) and the dynamics noise strengths (Q) for the "truth" model and the filter model were set equal to each other. The Q limits were determined by holding the R values constant while increasing the Q values one at a time. Each Q value was increased by orders of magnitude until the moving-bank's estimate no longer locked onto the true parameter. After the Q limits were found, the R limits were determined by holding the Q values constant and increasing the R values, one at a time, by orders of magnitude until the moving-bank's estimate of the true parameter no longer locked onto the actual true parameter. This degradation in performance was determined by maintaining a record of the true parameter estimate as in the two-state filter.

4.4.6. Controller Design: Need for Cross Terms in the Quadratic Cost

The goal of this study was to determine if the moving-bank controller performance could be increased by adding cross terms to the quadratic cost function. The cross weighting matrix (S) was evaluated [23:76] and compared to the state weighting matrix (X) and the control weighting matrix (U). The generalized cost function used

in this study is given in Equation (4-6) [23:73]. The differential equations used to solve for the cross terms are given in (4-7) through (4-12).

$$J = E\left\{\frac{1}{2}x^T(t_{N+1})X_f x(t_{N+1}) + \sum_{i=0}^N \left[\frac{1}{2}x^T(t_i)X(t_i)x(t_i) + u^T(t_i)U(t_i)u(t_i) + 2x^T(t_i)S(t_i)u(t_i)\right]\right\} \quad (4-6)$$

where

$x(t_i)$ = state vector

$u(t_i)$ = control input vector

$X(t_i)$ = state weighting matrix

$U(t_i)$ = control weighting matrix

$S(t_i)$ = cross term matrix

$$\dot{\Phi}(t, t_i) = F(t) \Phi(t, t_i) \quad (4-7)$$

$$\dot{\bar{B}}(t, t_i) = F(t) \bar{B}(t, t_i) + B(t) \quad (4-8)$$

$$\dot{\bar{X}}(t, t_i) = \Phi^T(t, t_i) W_{xx}(t) \Phi(t, t_i) \quad (4-9)$$

$$\begin{aligned} \dot{\bar{Q}}(t, t_i) = & F(t) \bar{Q}(t, t_i) + \bar{Q}(t, t_i) F^T(t) + \\ & G(t) Q(t) G^T(t) \end{aligned} \quad (4-10)$$

$$\dot{\bar{S}}(t, t_i) = \Phi^T(t, t_i) W_{xx}(t) \bar{B}(t, t_i) + \Phi^T(t, t_i) W_{xu}(t) \quad (4-11)$$

$$\begin{aligned} \dot{\bar{U}}(t, t_i) = & \bar{B}^T(t, t_i) W_{xx}(t) \bar{B}(t, t_i) + W_{uu}(t) \\ & + \bar{B}^T(t, t_i) W_{xu}(t) + W_{xu}^T(t) \bar{B}(t, t_i) \end{aligned} \quad (4-12)$$

The state transition matrix, Φ , is initially set equal to the identity matrix, while all the other initial conditions are set equal to a properly dimensioned zero matrix. Also the W_{xu} matrix was assumed to be equal to 0. These differential equations were solved using MACSYMA. In addition, because the matrices were of different dimensions, SS^T was formed and then the square root of the eigenvalues of the SS^T matrix were used to provide a comparison with the eigenvalues of the X matrix. During this study, it was determined that the cross terms do not significantly contribute to the generalized cost function. The detailed results of this study are contained in Section 5.7.

4.4.7. State Weighting and Control Weighting Matrix Tuning. The object of this study was to determine appropriate values for the X and U matrices. The preprocessor again produced six-state filters, while the primary processor used a single filter bank and was modified to output the true state vector. The true parameter and the filter parameter were both set to (7,6). The parameter point (7,6) was chosen so that it was near the center of the bank but not equal to a parameter point corresponding to one of the fixed-bank filters [(1,1), (1,5), (1,9), (5,1), (5,5), (5,9), (9,1), (9,5), (9,9)]. The state weighting matrix (X) values were determined by

holding the control weighting matrix (U) constant and increasing the X values one at a time until the rms values for the true states stop decreasing drastically as the X values were increased. After the X values were determined, the U values were found by holding the X values constant and decreasing the U values until the rms values for the true states stop decreasing drastically as the U values were decreased.

4.4.8. Controller Study. The goal of the controller study was to compare the performance of a moving-bank controller with a fixed-bank controller and a benchmark controller. The preprocessor again produced six-state filters, while the parameter space used in the primary processor consisted of the entire 100 points and the primary processor was modified to produce the true state vector. The postprocessor produced the true state vector and control vector statistics plots. This setup was used for the moving-bank controller and the fixed-bank controller. However in producing the benchmark controller, the filter parameter and true parameter values were set equal to each other and the parameter space used in the primary processor consists of one (true) point upon which to base one Kalman filter. Thus the optimal control was produced because the filter knows the true parameter. The comparison deals mainly with each controller's ability to quell oscillations.

4.5. Summary

The analysis technique used in this thesis is described in Section 4.2. In Section 4.3, the simulations themselves are described. Finally, the simulation plan used in this thesis is detailed in Section 4.4.

V. Results

5.1. Introduction

This chapter presents the results of the Monte Carlo simulations. The goal of this research effort was to ascertain the cause behind the poor performance of the moving bank in Karnick's research [9] and then to determine the optimal moving-bank or fixed-bank structure in order to provide the best control of the large space structure model. The first step was to find the dynamics noise strength (Q) and the measurement noise covariance (R) that provided for low rms error on the state variables, while not allowing the Q and R values to decrease to zero: a point of diminishing returns was subjectively established. Recall from Section 4.4.1, that this was accomplished by setting the "truth" and filter Q and R values equal to each other. These Q and R values were then varied one at a time. With these appropriate Q and R values, the center of the moving bank was pulled in the direction of the true parameter, rather than wandering aimlessly in parameter space. The system complexity was then increased to six states to determine if the performance of the moving bank decreased as the system complexity increased. From this study, it was determined that the performance of the moving bank did not degrade as system complexity increased.

Therefore, the problem behind the poor performance of the moving bank was the choice of the Q and R values: there is a fundamental requirement on the range of the Q and R values above which the adaptive mechanization of the moving bank breaks down. Section 5.6 discusses the range requirements on the Q and R values for this model.

However, even though the moving bank supplied an accurate estimate of the true parameter, some of the state estimation error plots were unstable. In an attempt to correct this problem a space discretization study was performed. After the completion of the space discretization study, the state estimation error plots were then stable (see Section 5.5). Subsequently, a study was conducted to determine if additional performance could be attained by adding cross terms, between states and controls, to the quadratic cost function used in the controller synthesis. Section 5.7 shows that the addition of cross terms to the quadratic cost function did not appreciably enhance the controller performance. After considering cross terms, additional performance was achieved by tuning the state weighting matrix (X) and the control weighting matrix (U). This tuning was performed for the (7,6) parameter point, which was chosen so that the tuned parameter point would be near the center of the bank

but not equal to a parameter value corresponding to one of the fixed bank filters $[(1,1),(1,5),(1,9),(5,1),(5,5),(5,9),(9,1),(9,5),(9,9)]$. The results of this tuning process are given in Section 5.8. The final attempt to enhance performance was the use of a transition period of partially adaptive control between the time used to estimate states in open loop fashion (with a dither signal to excite the system) and the time when full multiple model adaptive control was applied to the system. It was thought the transition period was needed because if adaptive control inputs were supplied to the system before an accurate estimate of the true parameter was obtained, the control inputs could drive the system unstable. Therefore, a nonadaptive controller was designed that was very robust to parameter variation at the expense of controller performance so that the controller inputs would at least maintain stability regardless of the estimate of the true parameter. However, Section 5.9 shows that this transition period was not needed because the moving-bank algorithm locks onto the true parameter in less than 0.5 sec, thereby allowing full adaptive control to be applied to the system immediately following the excitation and open loop estimation period. It was also determined during this controller evaluation, the state weighting matrix (X) and the control weighting matrix (U) needed to be tuned for each parameter point

in order to achieve the best performance. This is also covered in Section 5.9.

Finally after all the techniques used to enhance performance were applied to the system, a performance comparison between a benchmark controller, a moving-bank controller, and a fixed-bank controller was conducted, with the results shown in Section 5.9. This performance comparison showed that the moving-bank algorithm converges to the performance of the benchmark controller, while the fixed-bank controller converges to the benchmark performance only when the true parameter corresponds to one of the filters implemented in the fixed-bank controller. The benchmark controller is described in Section 5.9 and is considered to provide the optimal control performance.

5.2. Measurement Covariance Noise and Dynamics Noise Strength Determination

Before the research can be started, appropriate choices of the measurement noise covariance (R) and dynamics noise strength (Q) must be found. A description of the process used to determine appropriate choices of the R and Q values is given in Section 4.4.1. Recall though that the truth and filter-assumed values were set equal to each other, then both the truth and filter values for the entries in these matrices are varied one at a time. Also

note, that the measurement noise covariance (R) and dynamics noise strength (Q) are determined using a single filter primary processor. The resulting Q and R values obtained from this process are given in Table 5-1. Table 5-2 contains the time averaged state estimation error statistics, omitting the initial transient time, for the Q and R values given in Table 5-1. These statistics were also produced using Q and R values which were one order of magnitude larger than in Table 5-1. These time averaged statistics are contained in Table 5-3. As can be seen, the rms errors in Table 5-3 are about one order of magnitude larger than those in Table 5-2. Therefore, the Q and R values contained in Table 5-1 provide better estimation performance than the Q and R values used in Table 5-3. After appropriate Q and R values are determined, the remainder of the research effort can be performed.

5.3. Two-State Model

The goal of the two-state model was to determine if Hentz's work [6] could be duplicated for the large space-structure model. Hentz showed that, for a two-state (i.e. one bending mode) filter model matched against a two-state truth model, the center of the moving bank would move in the direction of the true parameter. However, it is important to note that optimal performance of the

Table 5-1

Dynamics and Measurement Noise Strength
(for both the truth and filter models)

Dynamics Noise Strengths

(Units: Position - in^2/sec ; Velocity - in^2/sec^3)

Noise Input at Node 1 (Q1) - 80.

Noise Input at Node 2 (Q2) - 80.

Noise Input at Hub (Q3) - 50.
at angular velocity
level

Noise Input at Hub (Q4) - 50.
at angular
acceleration level

Measurement Noise Variances

(Units: Position - in^2/sec ; Velocity - in^2/sec)

Position Sensor at Node 1 (R1) - 0.000027

Velocity Sensor at Node 1 (R3) - 0.0025

Position Sensor at Node 2 (R2) - 0.0000027

Velocity Sensor at Node 2 (R4) - 0.0042668

Position Sensor at Hub (R5) - 12.0

Velocity Sensor at Hub (R6) - 0.000855

Table 5-2

Time Averaged State Estimation Errors Statistics:
 Using the Q and R Values Given in Table 5-1;
 For Parameter Point (5,5)
 (Units: Position - in; Velocity - in/sec)

State	Mean	Standard Deviation
Position Error (Node 1)	0.008639	0.002310
Velocity Error (Node 1)	0.05540	0.006272
Position Error (Node 2)	0.01925	0.0008319
Velocity Error (Node 2)	0.03963	0.046079
Position Error (Hub)	-0.21341	0.43714
Velocity Error (Hub)	0.9989	3.3993

Table 5-3

Time Averaged State Estimation Errors Statistics:
 Using the Q and R Values One Order of Magnitude Larger
 than in Table 5-1;
 For Parameter Point (5,5)
 (Units: Position - in; Velocity - in/sec)

State	Mean	Standard Deviation
Position Error (Node 1)	0.0202	0.023
Velocity Error (Node 1)	0.172	0.06284
Position Error (Node 2)	0.0657	0.00822
Velocity Error (Node 2)	0.12559	0.4607
Position Error (Hub)	-0.673	4.3713
Velocity Error (Hub)	3.1657	33.993

moving bank was not a goal of this study. Recall from Section 4.4.2, that both the truth and filter models were reduced to two states (i.e., one bending mode). In the Monte Carlo simulations, a dither signal (square wave) with a magnitude of 300 and frequency of 30 rad/sec was used to excite the system. The magnitude and frequency of the dither signal were chosen by trial and error using the criteria that the system be excited but not dominated by the dither signal. In order to determine the trends in the movement of the bank, a record was needed of the estimated true parameter [9]. This record was acquired by initializing a 10x10 matrix of elements to zero and then adding one at each sample time to the matrix element corresponding to the estimate of the true parameter. Entries were recorded from $t = 1.0$ to the end of the run. Recording was not started at $t = 0.0$ in an attempt to avoid transient effects. The transient effects could be omitted because they were short lived and stable. This is shown in Section 5.9. In addition, the moving-bank algorithm used the smallest discretization level and was initially centered at the (5,5) parameter point (i.e., in the middle of the parameter space). Using the above techniques, data for Figure 5-1 through Figure 5-3 was acquired for three true parameter points. In Figure 5-1, the estimated true parameter was spread out somewhat in parameter space. However, most

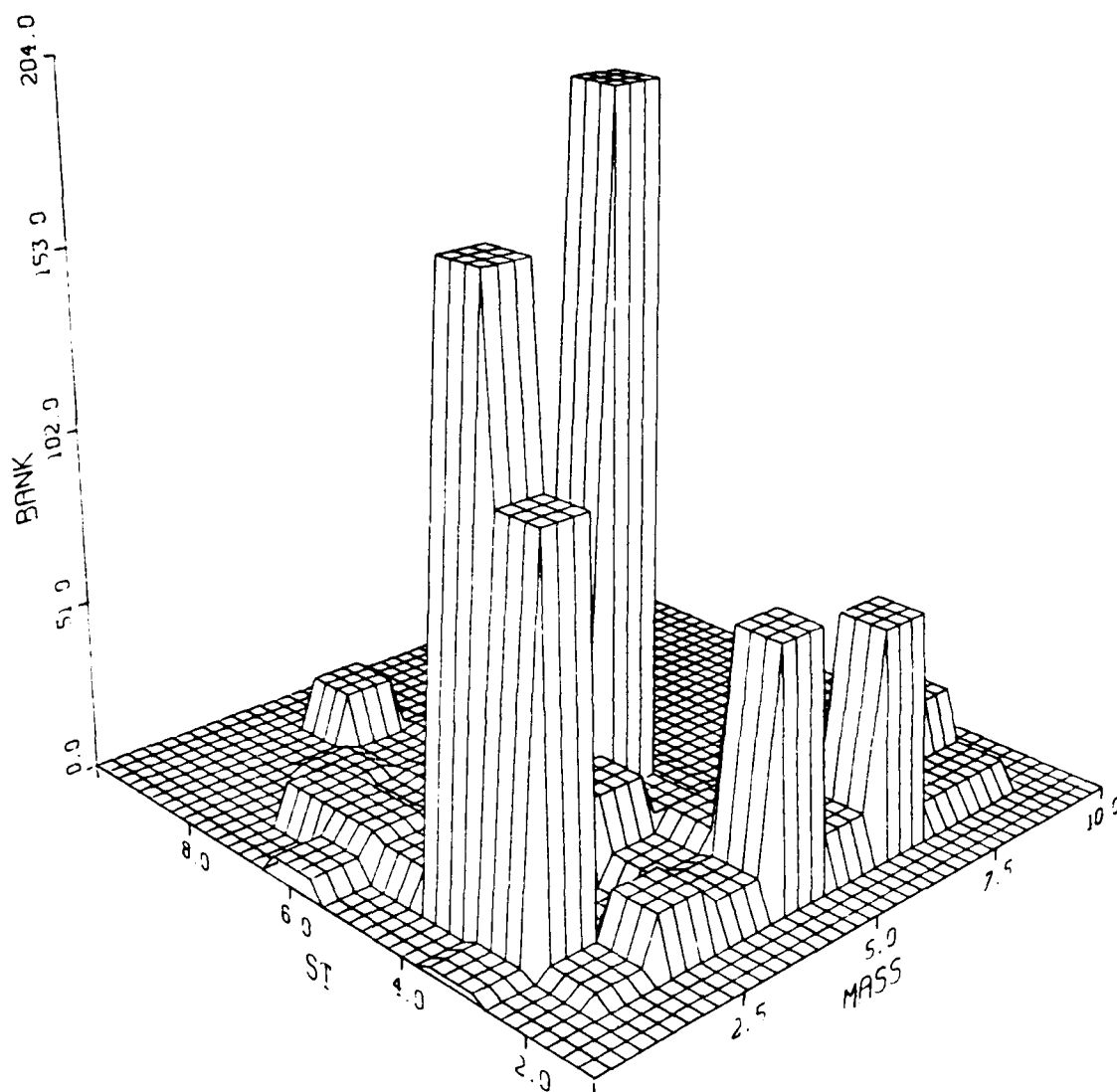


Fig. 5-1. Two State Model Bank Location Time History;
True Parameter at Mass = 1, Stiffness = 1

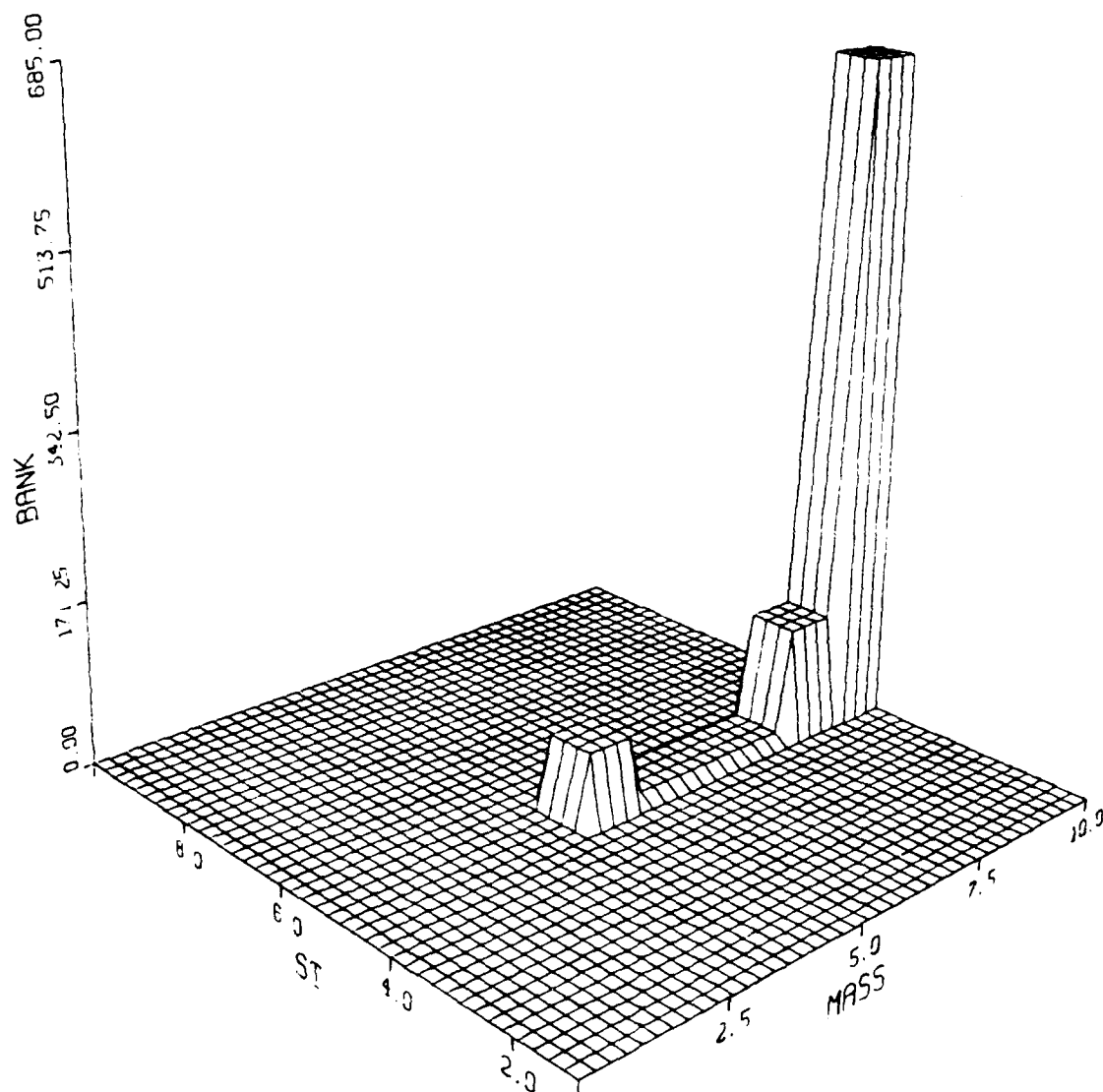


Fig. 5-2. Two-State Model Bank Location Time History;
True Parameter at Mass = 9, Stiffness = 2

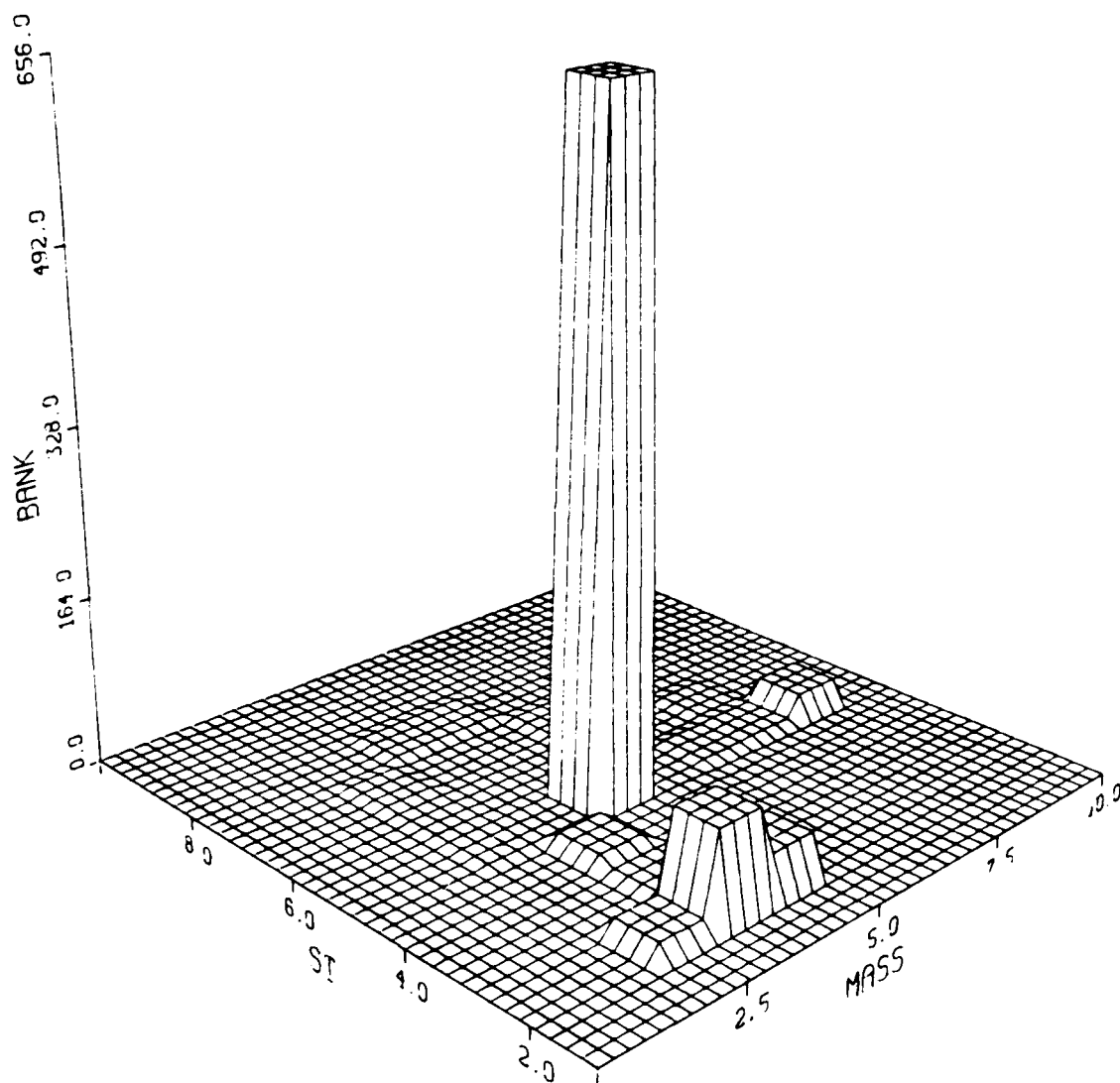


Fig. 5-3. Two-State Model Bank Location Time History;
True Parameter at Mass = 2, Stiffness = 9

of the data points were contained in the lower left quadrant which indicates a pulling of the center of the moving bank in the correct direction. In Figure 5-2, the mass parameter is pulled in the correct direction but the stiffness parameter is not pulled in any direction. Again in Figure 5-3, the mass parameter is pulled in the correct direction while this time the stiffness parameter is actually pulled in the wrong direction. However, there is an overall indication of a pulling trend of the center of the moving bank toward the true parameter. Therefore, Hentz's work was duplicated for this model. The quality of the state estimate produced by the moving-bank algorithm is given in the six-state filter study.

5.4. Six-State Filter

Since Hentz's work was duplicated for the one-bending mode model, it was desired to determine if the performance of the moving-bank algorithm degrades for a more complex system. Therefore, the state dimension of both the filter model and the "truth" model were increased to six (i.e., one rigid body mode and two bending body modes). Recall from Section 4.4.3, there was no purposeful order reduction in generating the filter model, and thus the robustness of the algorithm to such model mismatch has not as yet been explored. The data shown in Figures 5-4 through 5-7 was

acquired using the same techniques as the data used in Figures 5-1 in the two-state model study given in Section 5.3, except the model now has six states. Figures 5-4 through 5-7 show that there is a definite pulling of the center of the moving bank toward the true parameter point. In analyzing Figures 5-4 through 5-7 as compared to Figures 5-1 through 5-3, it appears that the six-state system performed better than the two-state system, which would indicate that the moving bank performed better as the system complexity increases. However, recall from Section 5.2, that the Q and R values were chosen for the six-state model. These Q and R values were used in the two-state model study, therefore the Q and R values were not tuned for the two-state model study. However, the goal of the two-state model study was to show the trend of the center of the bank, not to obtain optimal performance. Therefore the six-state model should outperform the two-state model. The performance of the six-state filter indicates that system complexity does not appreciably affect the performance of the moving-bank parameter estimation. However, the six-state model still has a performance problem. This problem is illustrated in Figures 5-8 through 5-10, which contain the state estimation error for the (2,7), (9,9), (7,6) parameter points. In these figures, the mean value

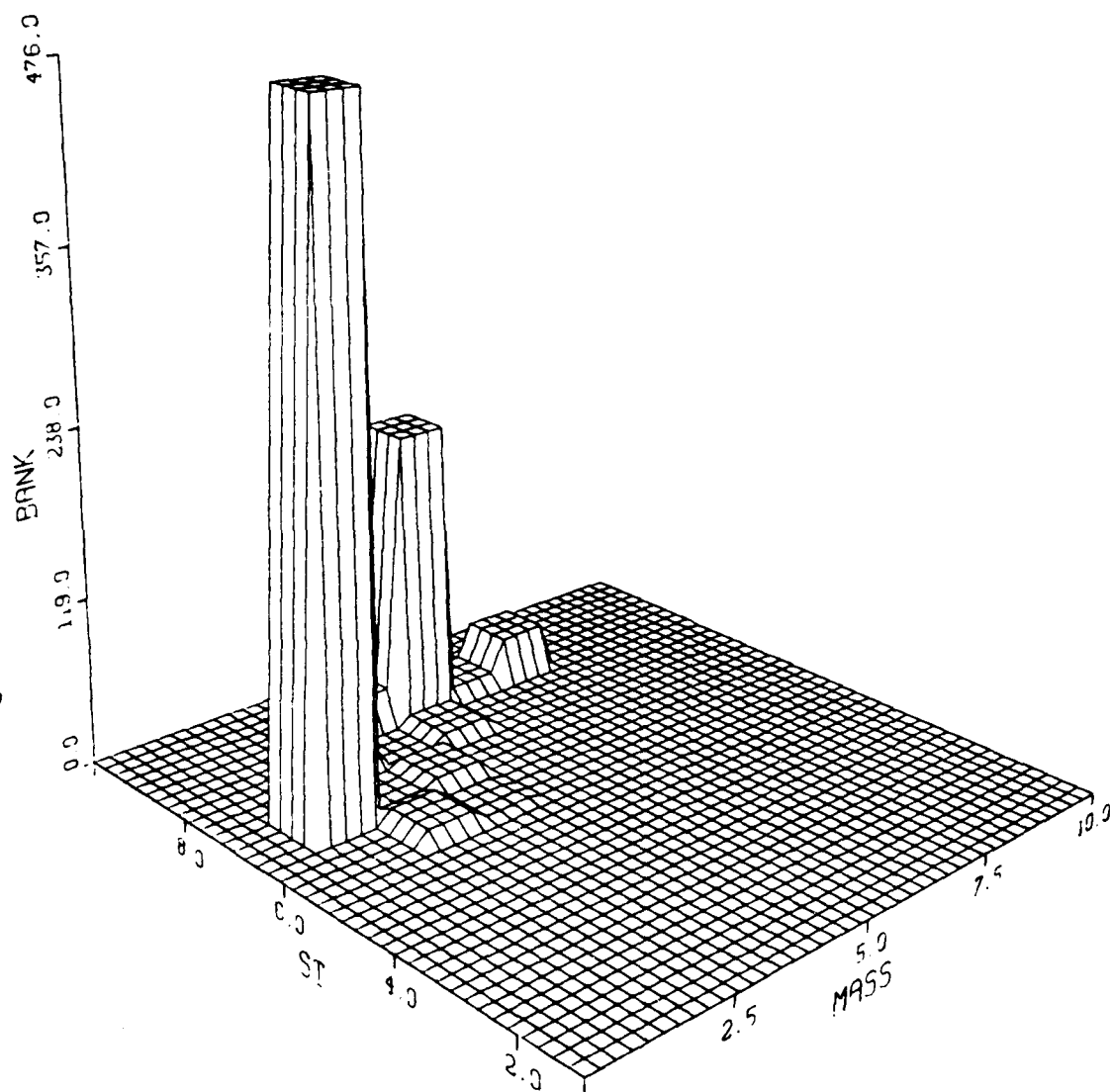


Fig. 5-4. Six-State Model Bank Location Time History;
True Parameter at Mass = 2, Stiffness = 7
Old Discretization Level

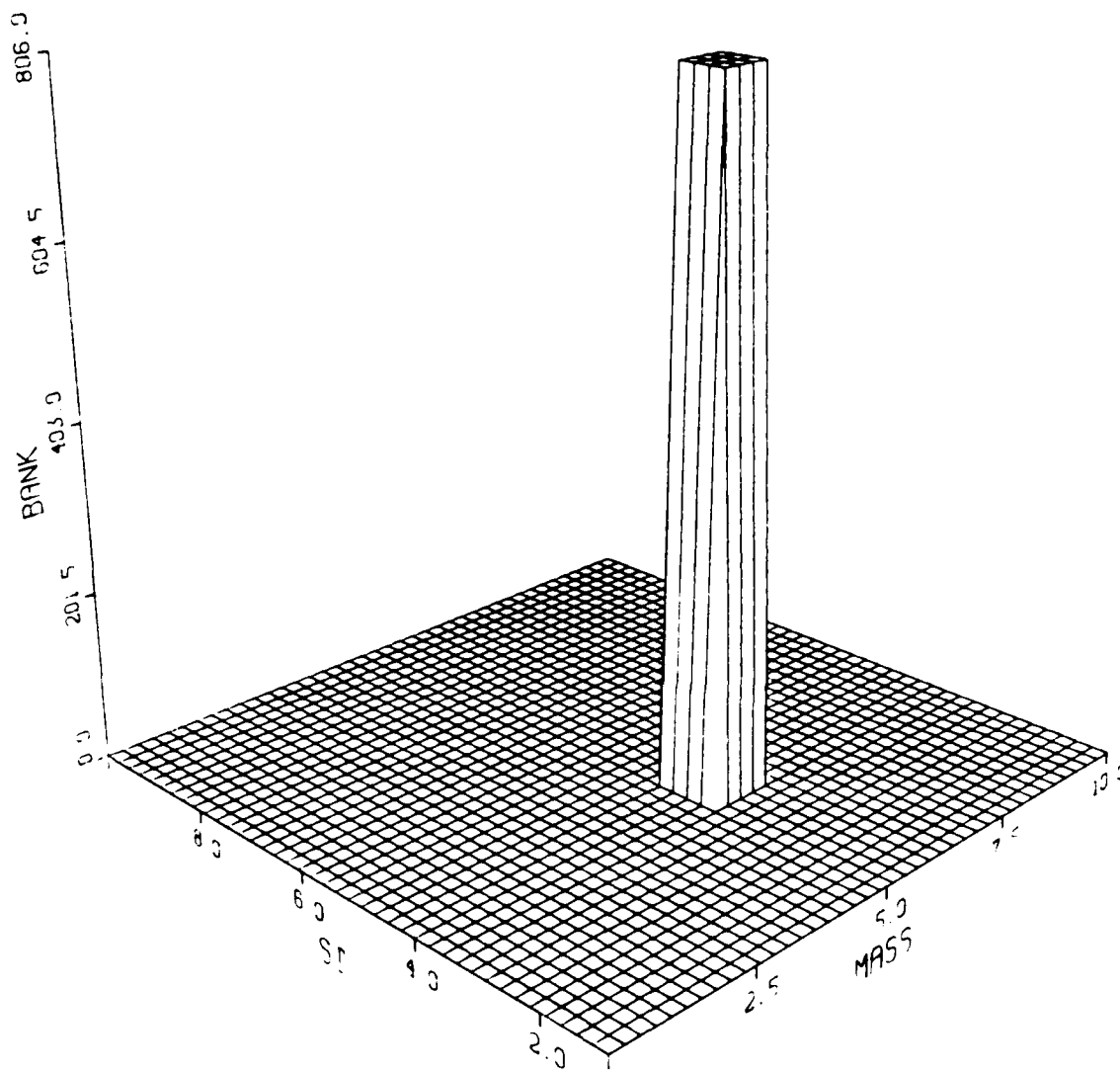


Fig. 5-5. Six-State Model Bank Location Time History;
True Parameter at Mass = 6, Stiffness = 4
Old Discretization Level

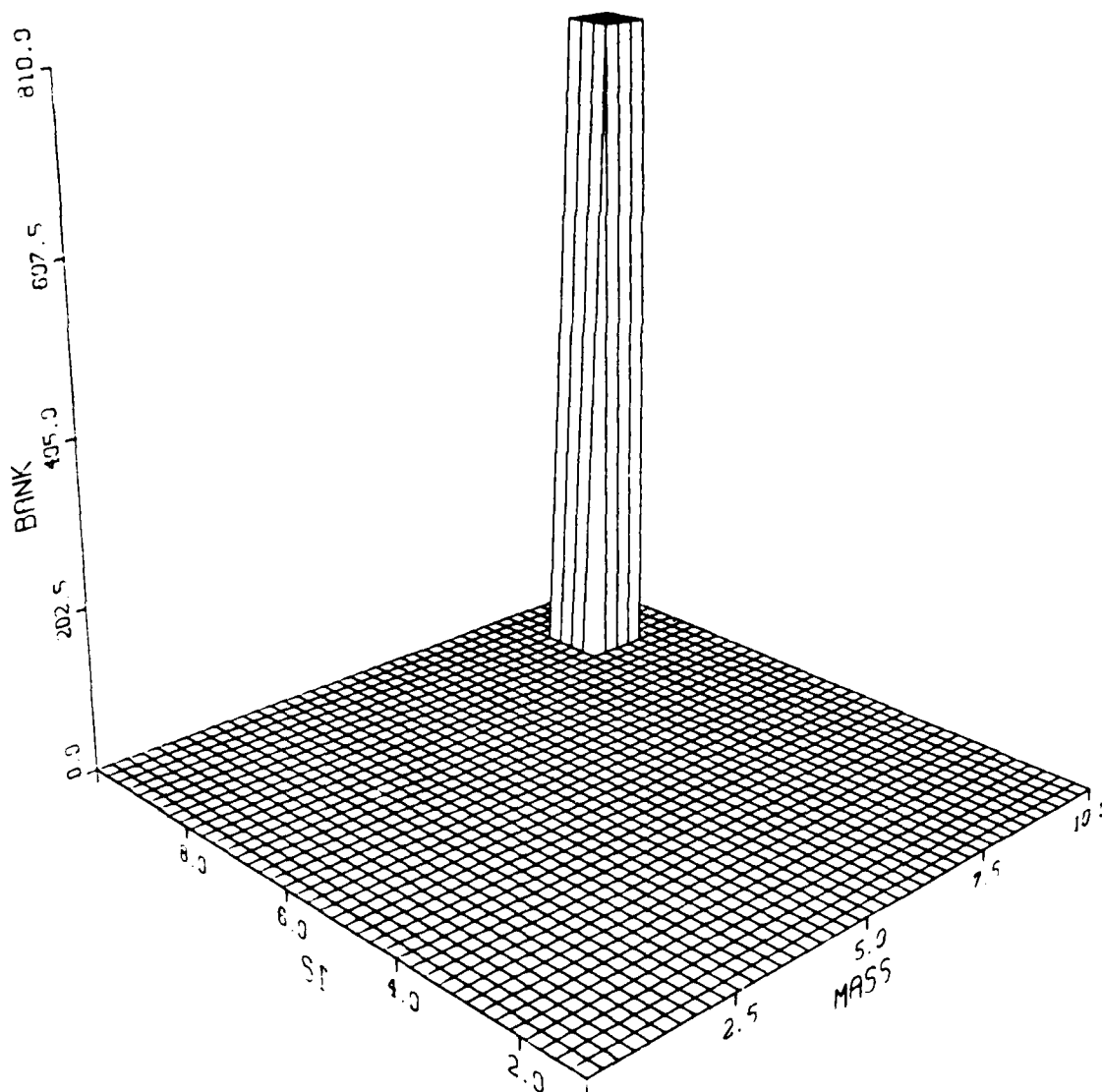


Fig. 5-6. Six-State Model Bank Location Time History;
True Parameter at Mass = 9, Stiffness = 9
Old Discretization Level

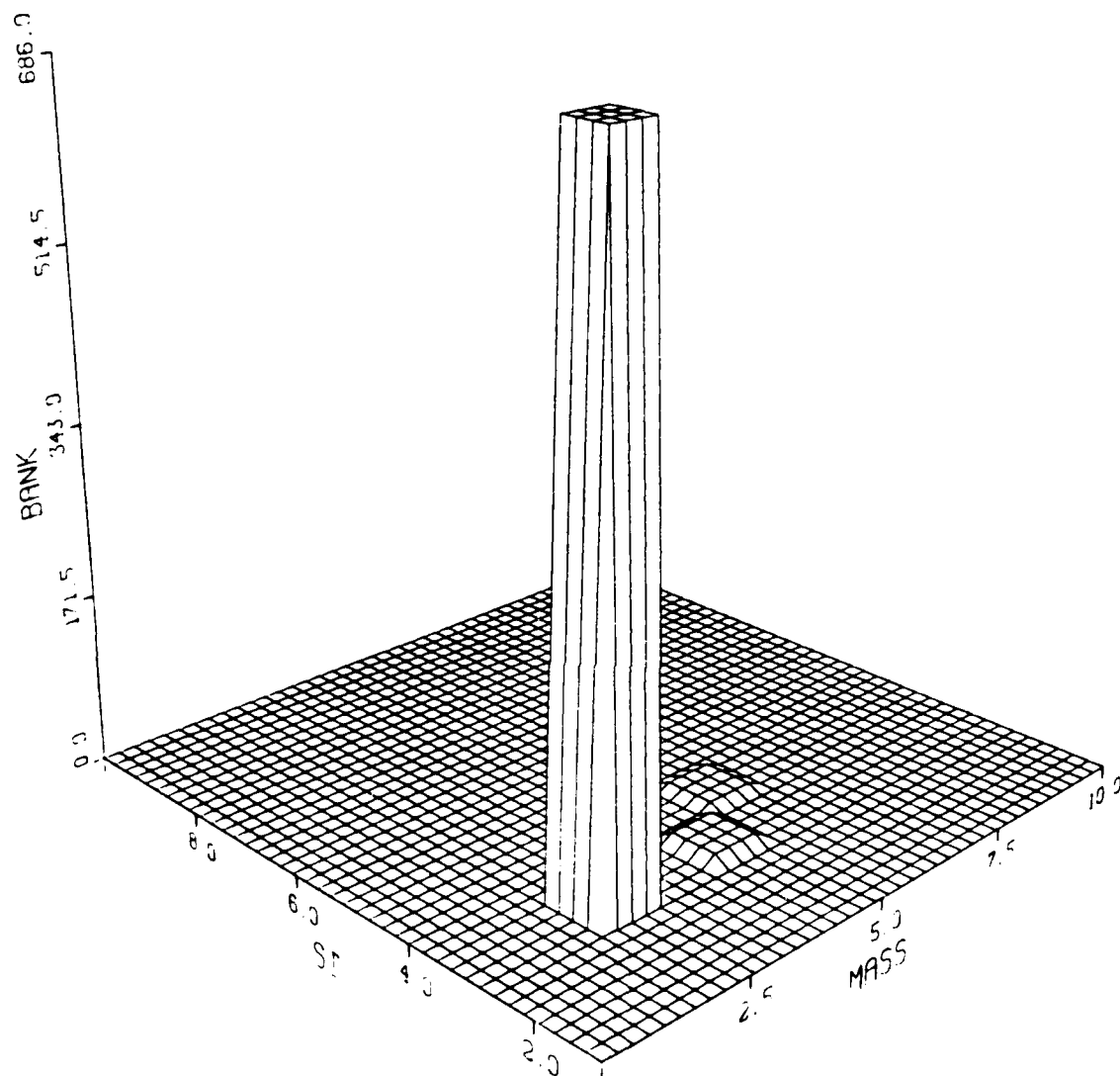


Fig. 5-7. Six-State Model Bank Location Time History;
 True Parameter at Mass = 3, Stiffness = 3
 Old Discretization Level

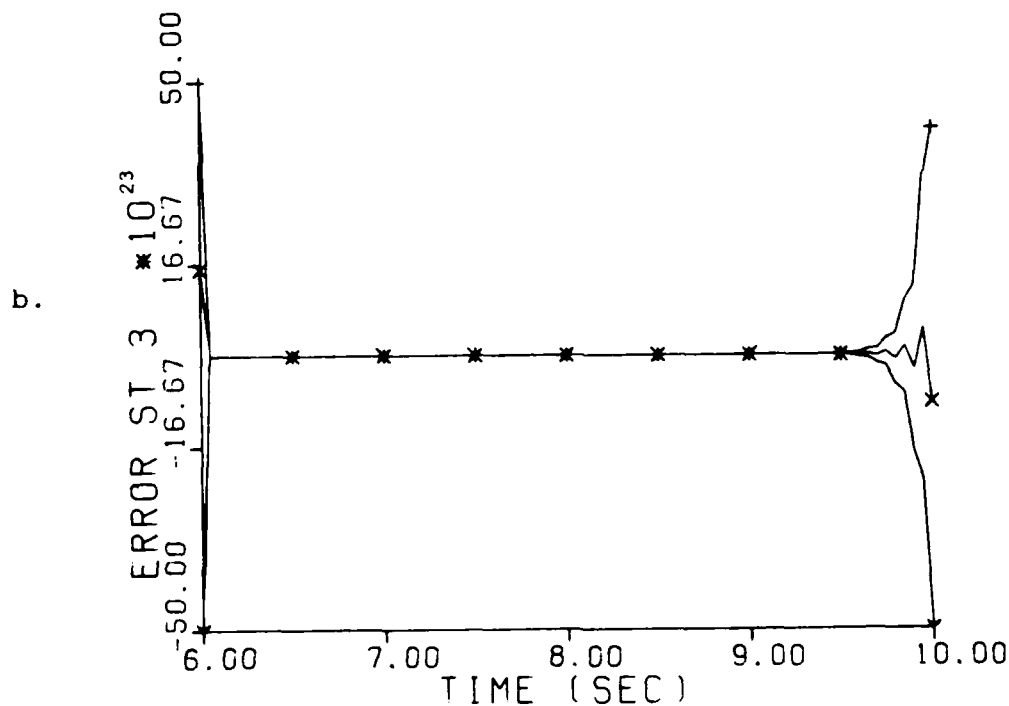
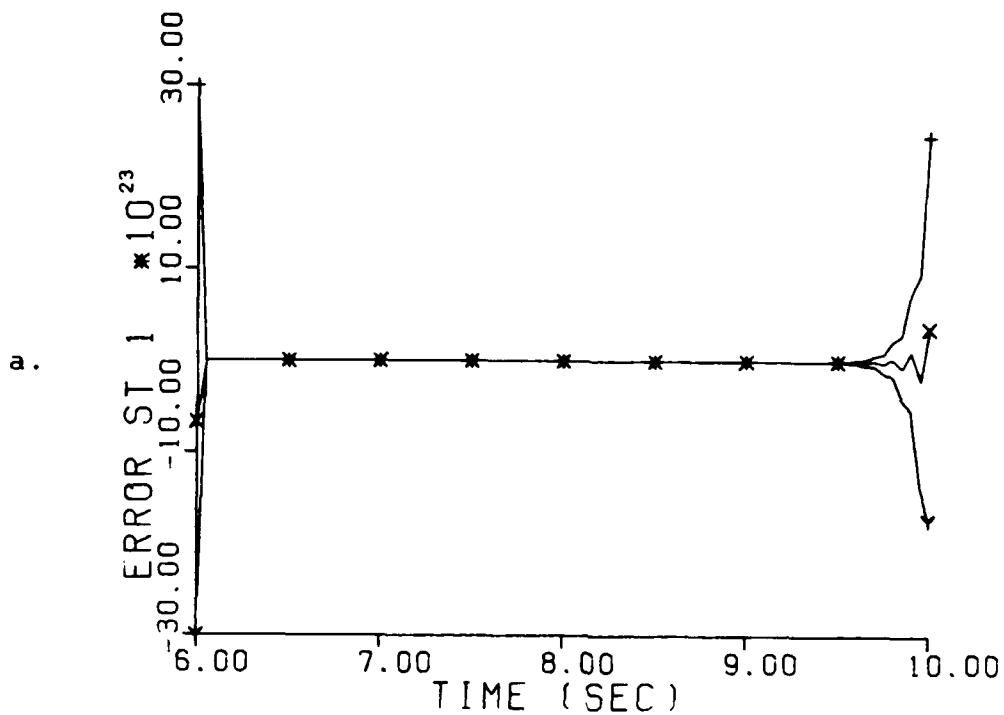


Fig. 5-8. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 Old Discretization Level
 a. Position Error at Actuator #1 (in)
 b. Velocity Error at Actuator #1 (in/sec)

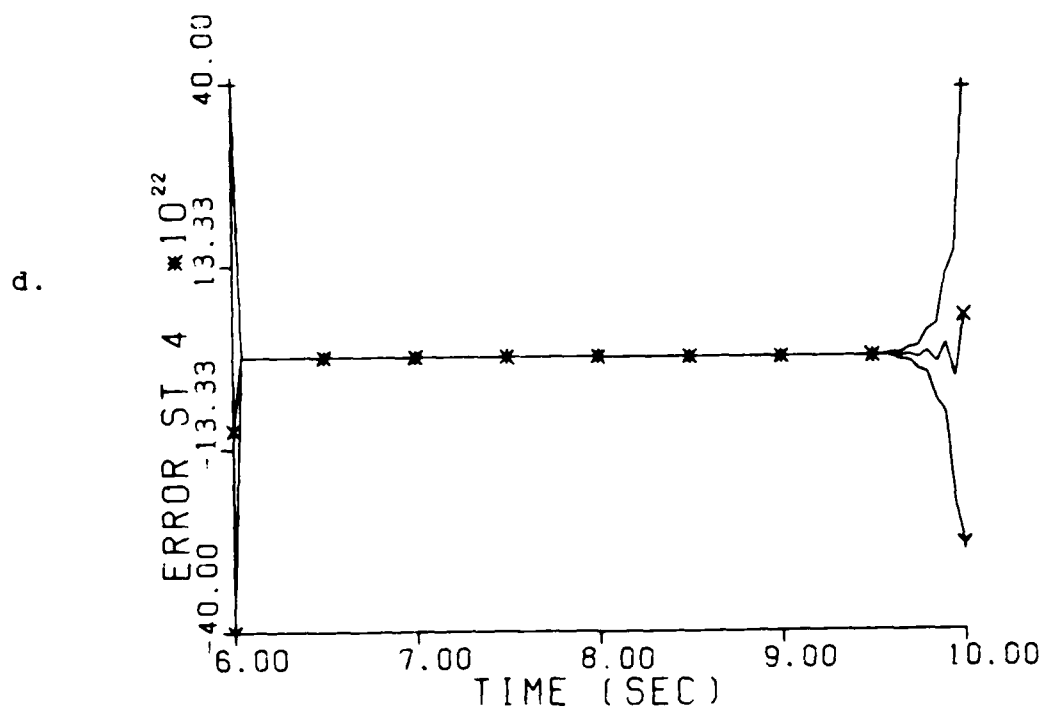
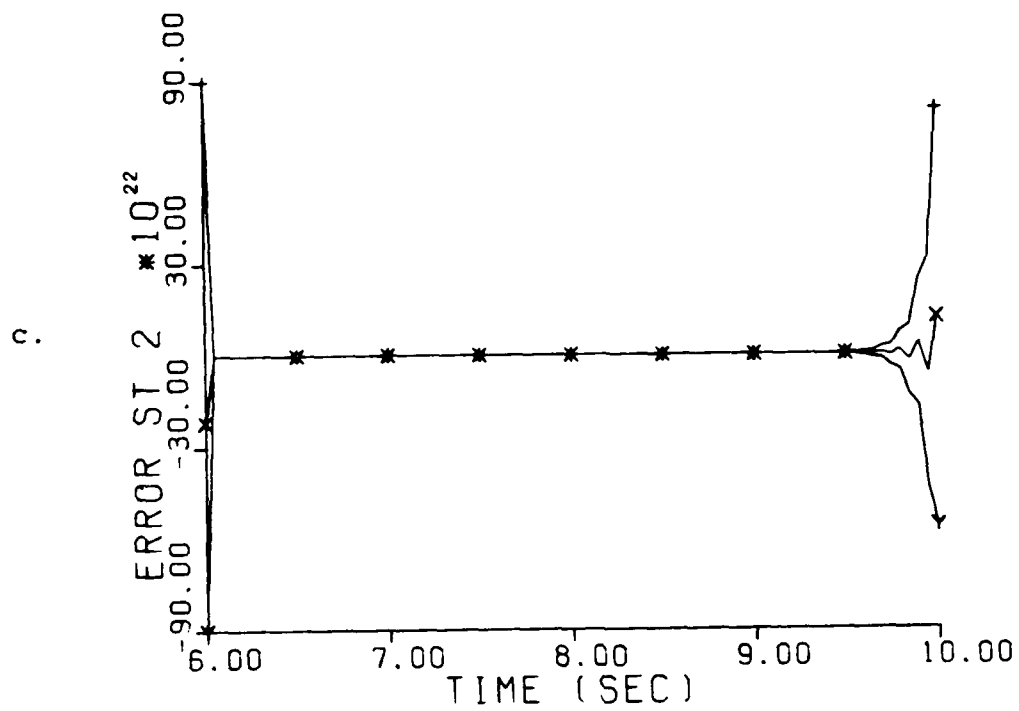


Fig. 5-8. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 Old Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

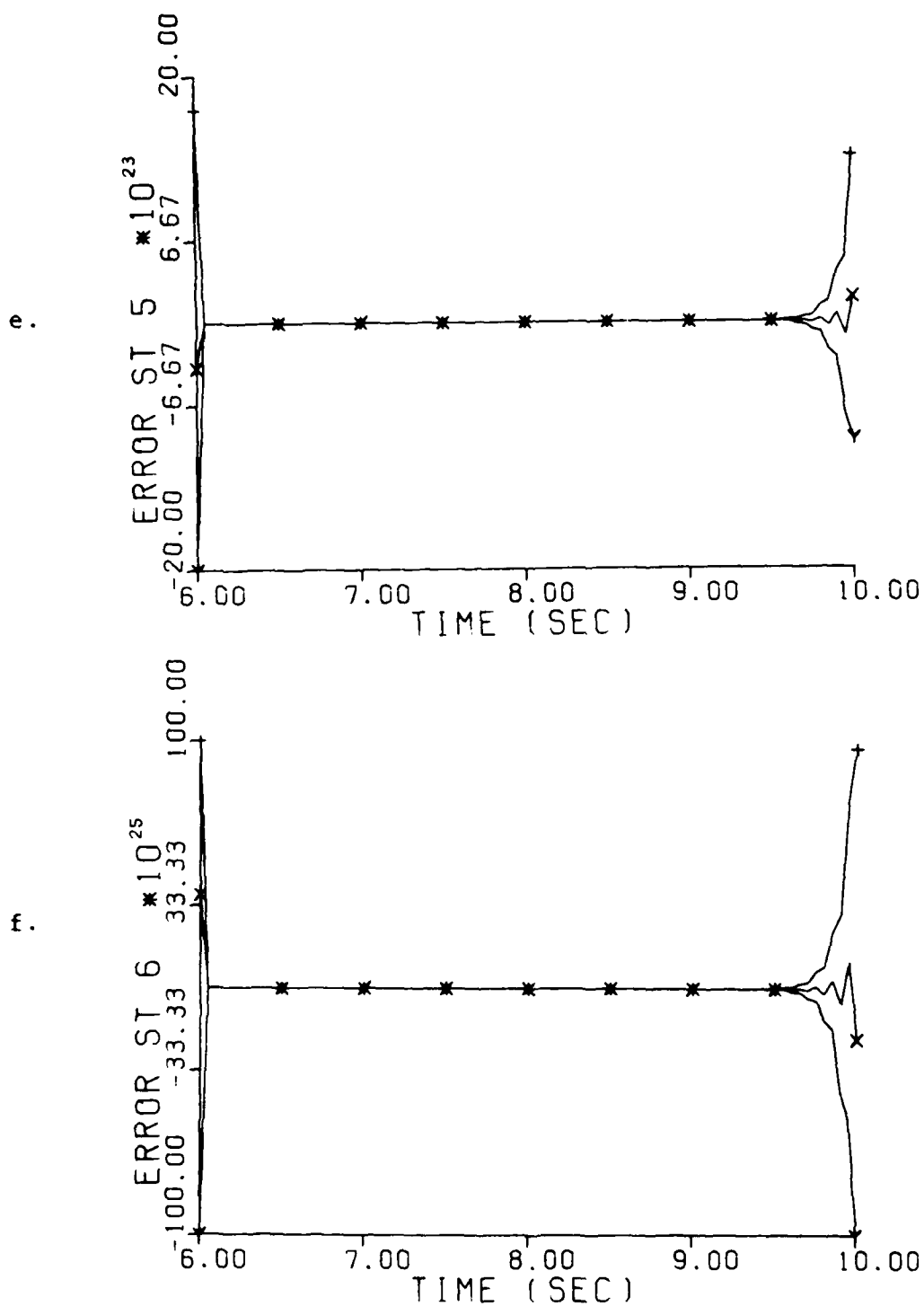


Fig. 5-8. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 Old Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

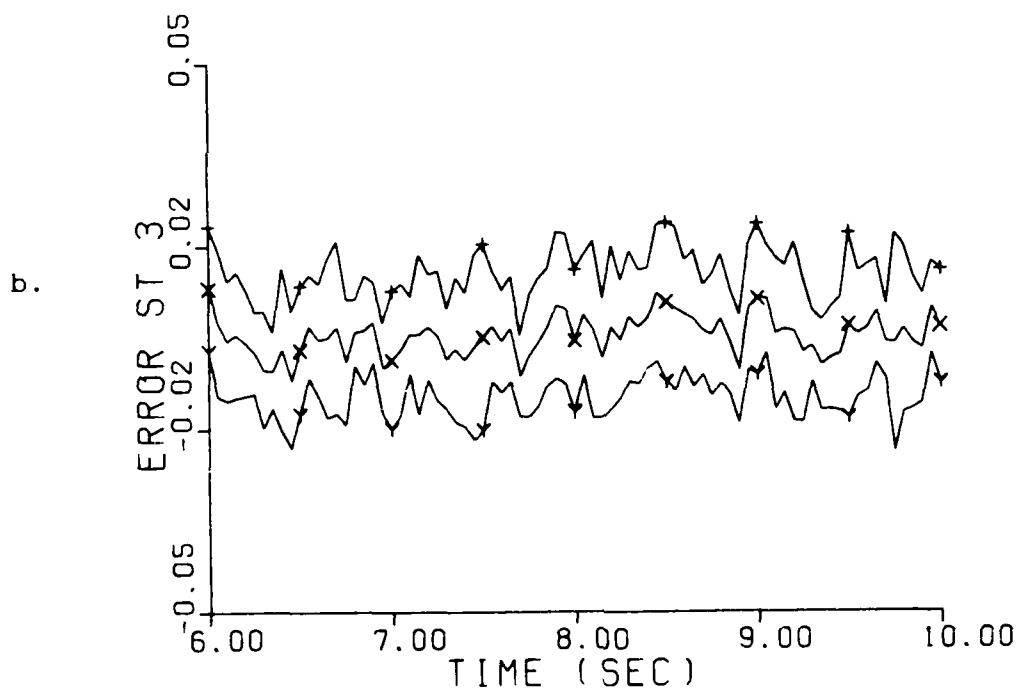
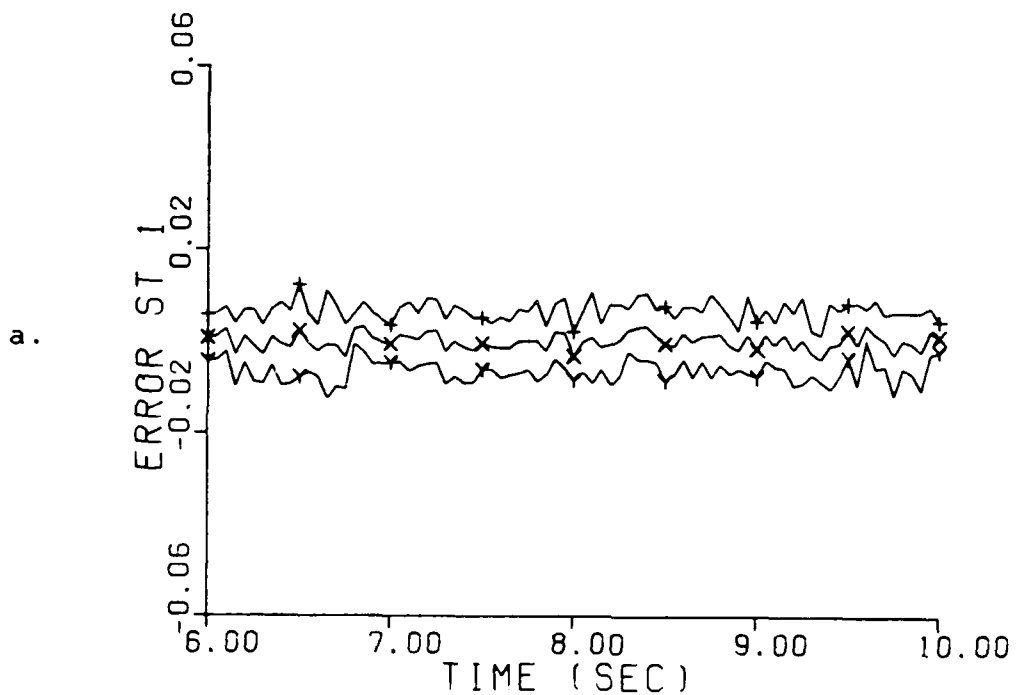


Fig. 5-9. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 9, Stiffness = 9
 Old Discretization Level
 a. Position Error at Actuator #1 (in)
 b. Velocity Error at Actuator #1 (in/sec)

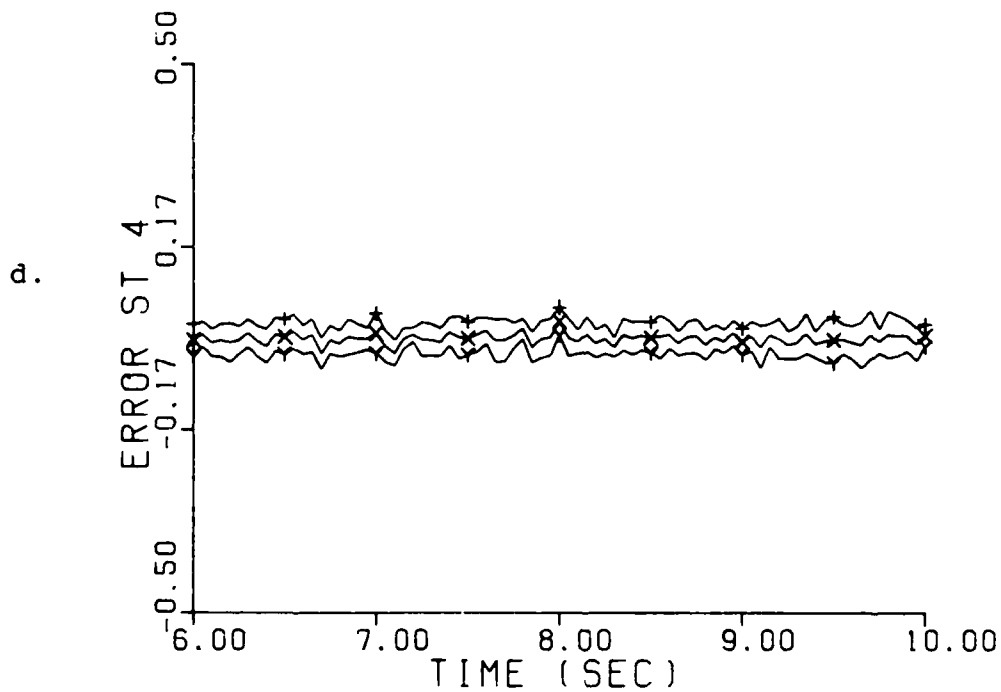
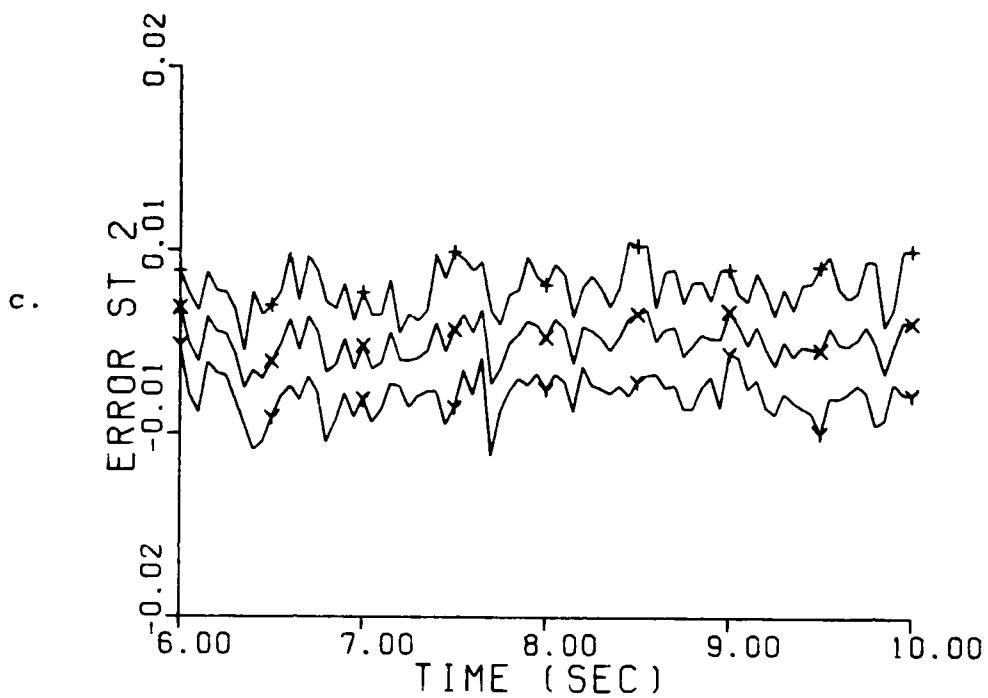


Fig. 5-9. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 9, Stiffness = 9
 Old Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

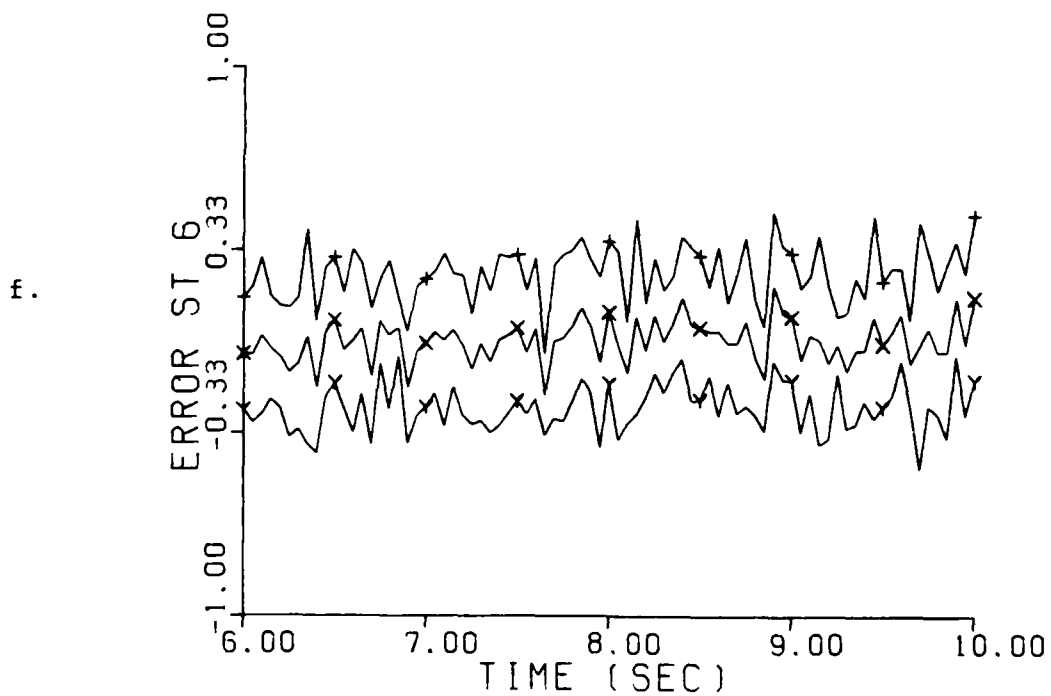
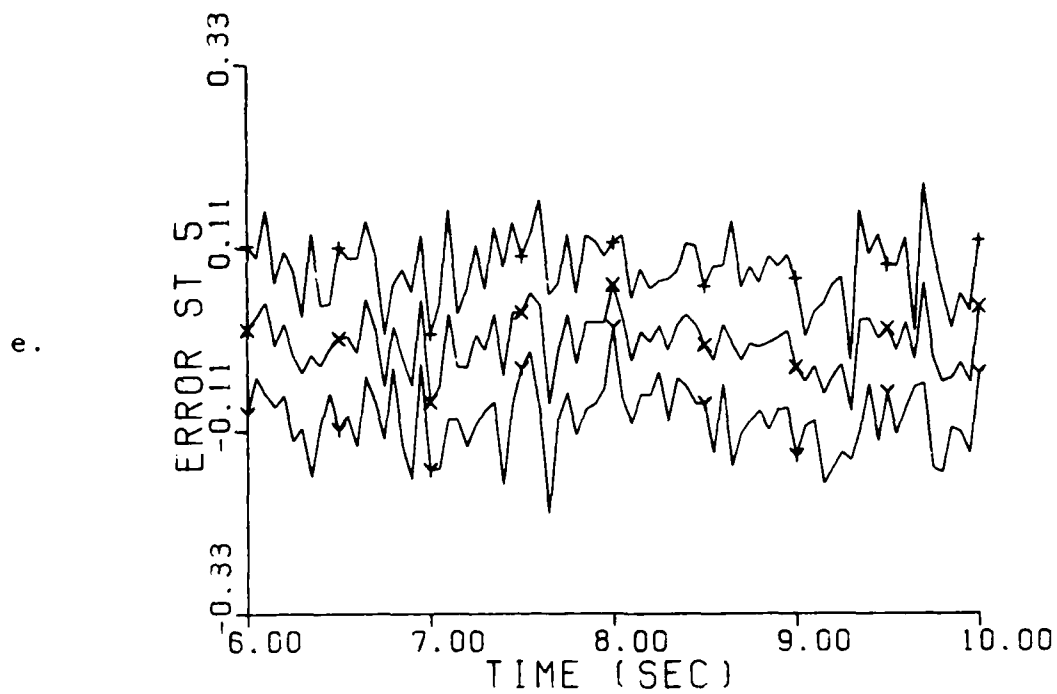


Fig. 5-9. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 9, Stiffness = 9
 Old Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

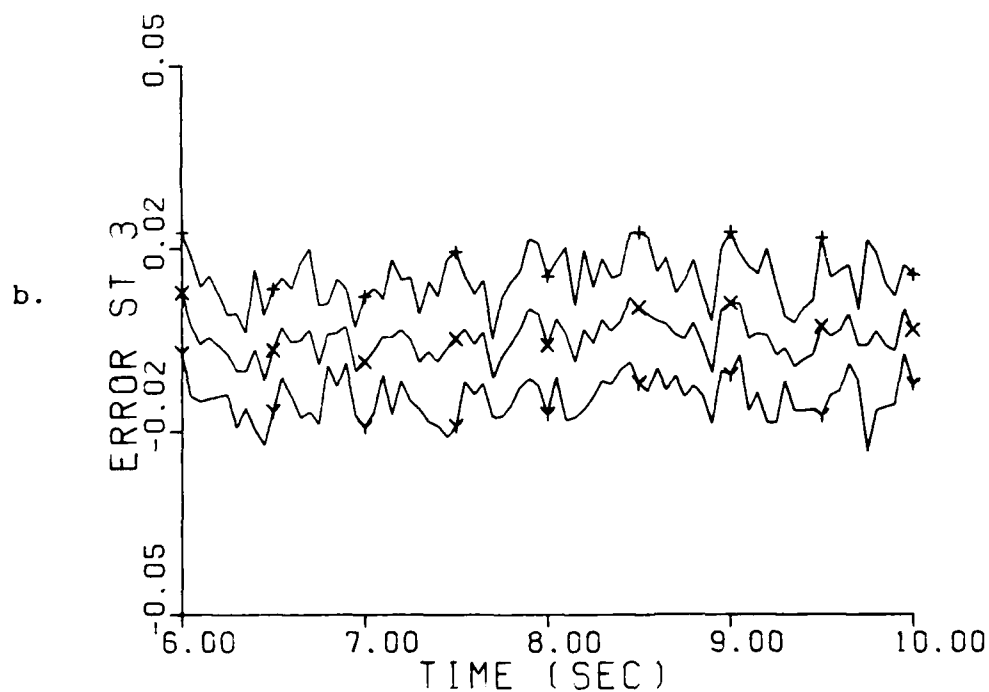
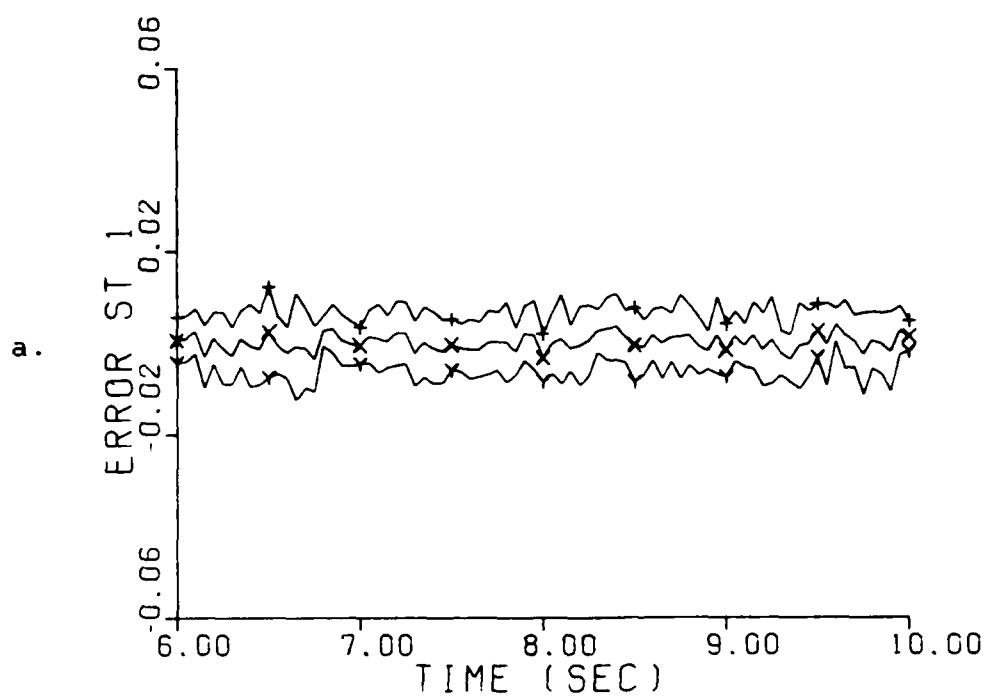


Fig. 5-10. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 Old Discretization Level
 a. Position Error at Actuator #1 (in)
 b. Velocity Error at Actuator #1 (in/sec)

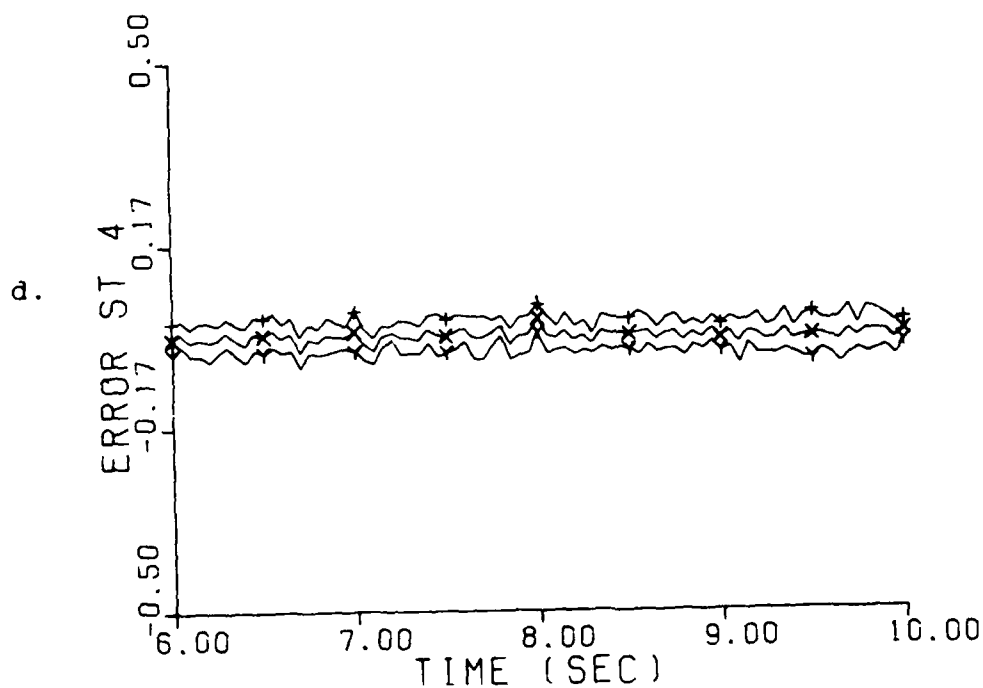
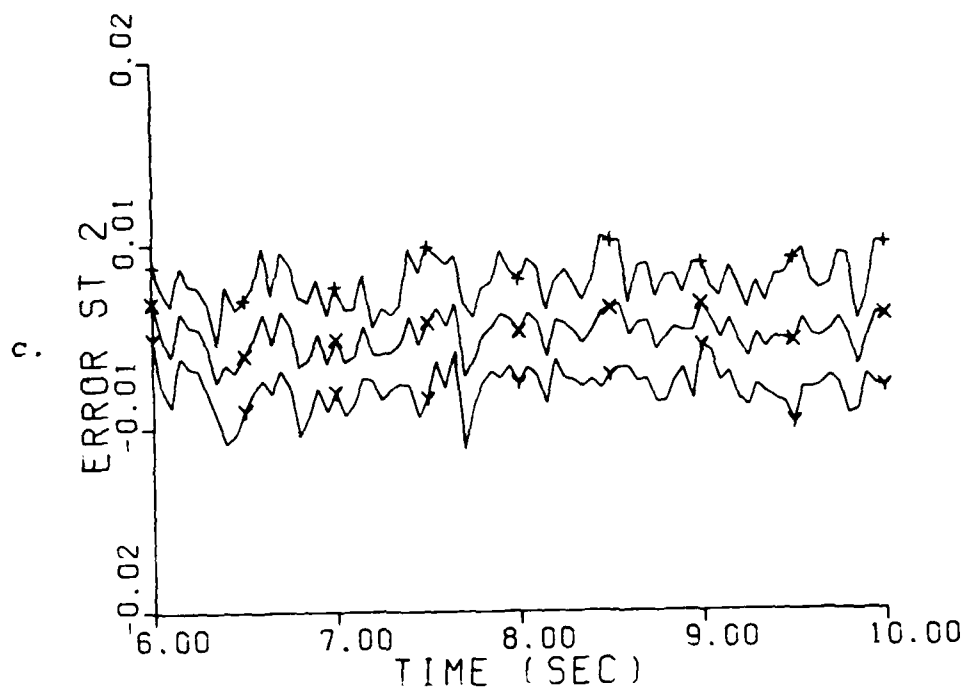


Fig. 5-10. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 Old Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

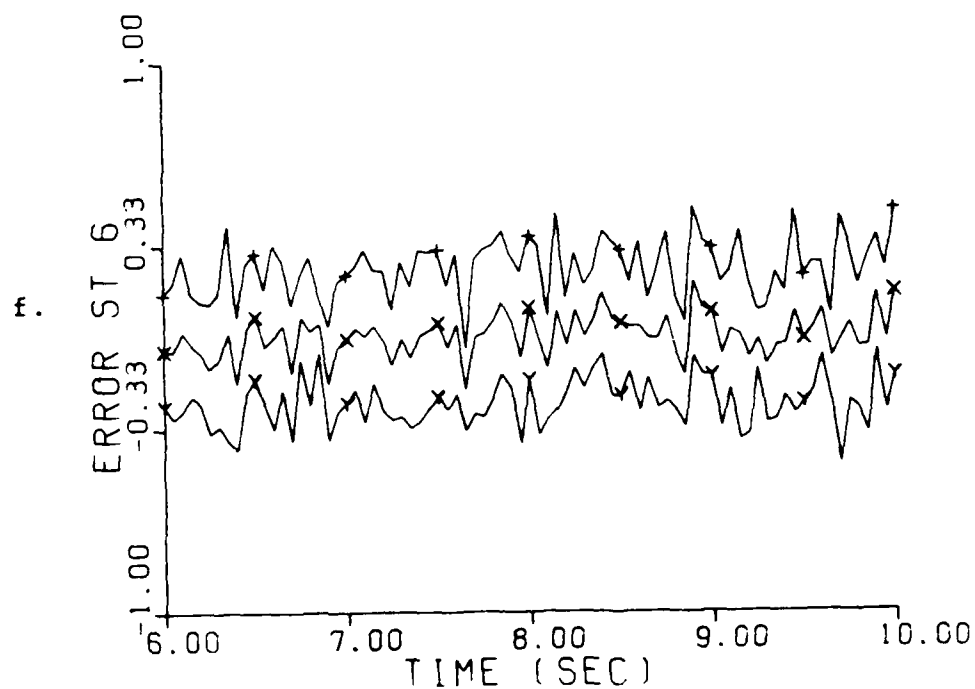
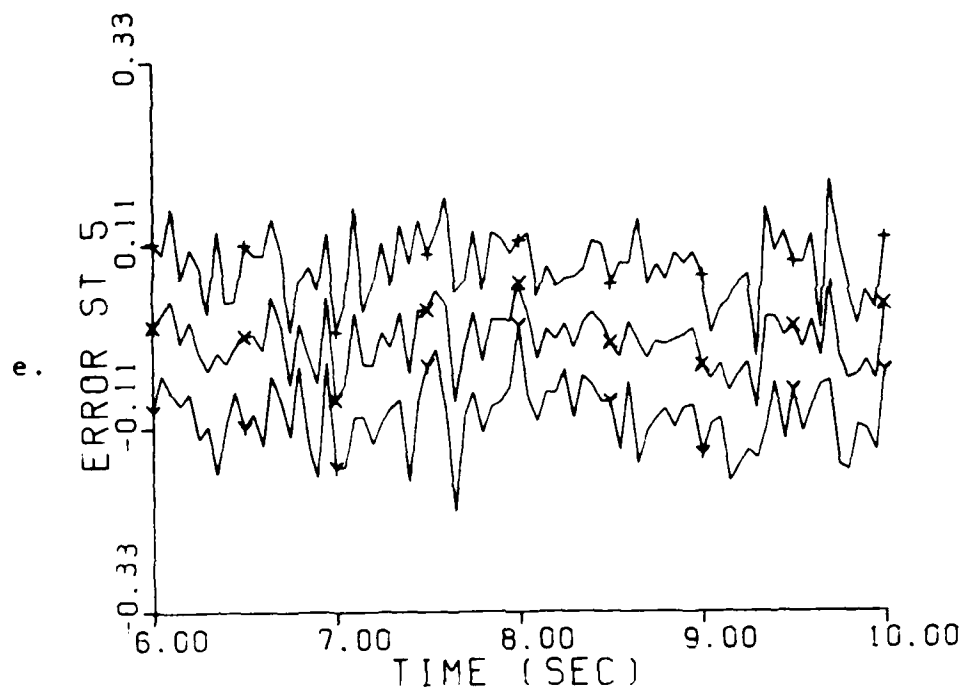


Fig. 5-10. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 Old Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

(symbol x), the mean plus one sigma (symbol $+$), and the mean minus one sigma (symbol y) are plotted for each of the six states in the filter. Recall in Figure 5-4, the moving-bank's estimate of the true parameter did not lock onto the actual true parameter value. The corresponding state estimation errors for the (2,7) parameter point are unstable (see Figure 5-8). However, recall in Figure 5-6, the moving bank's estimate of the true parameter did lock onto the actual true parameter value. The corresponding state estimate error plots for the (9,9) parameter point are stable (see Figure 5-9). This seems to indicate that there is a strong correlation between the ability of the moving-bank algorithm to estimate the true parameter and its ability to provide precise state estimates. Couple this correlation with the fact that the old space discretization level (see Figure 5-11) was determined by intuition. This indicates that a space discretization level study must be performed to determine an appropriate space discretization level.

5.5. Space Discretization

Recall from Section 4.4.4, the space discretization study was accomplished by monitoring the rms error of the state variables as the true parameter and the filter parameter are moved apart. The results of this study are

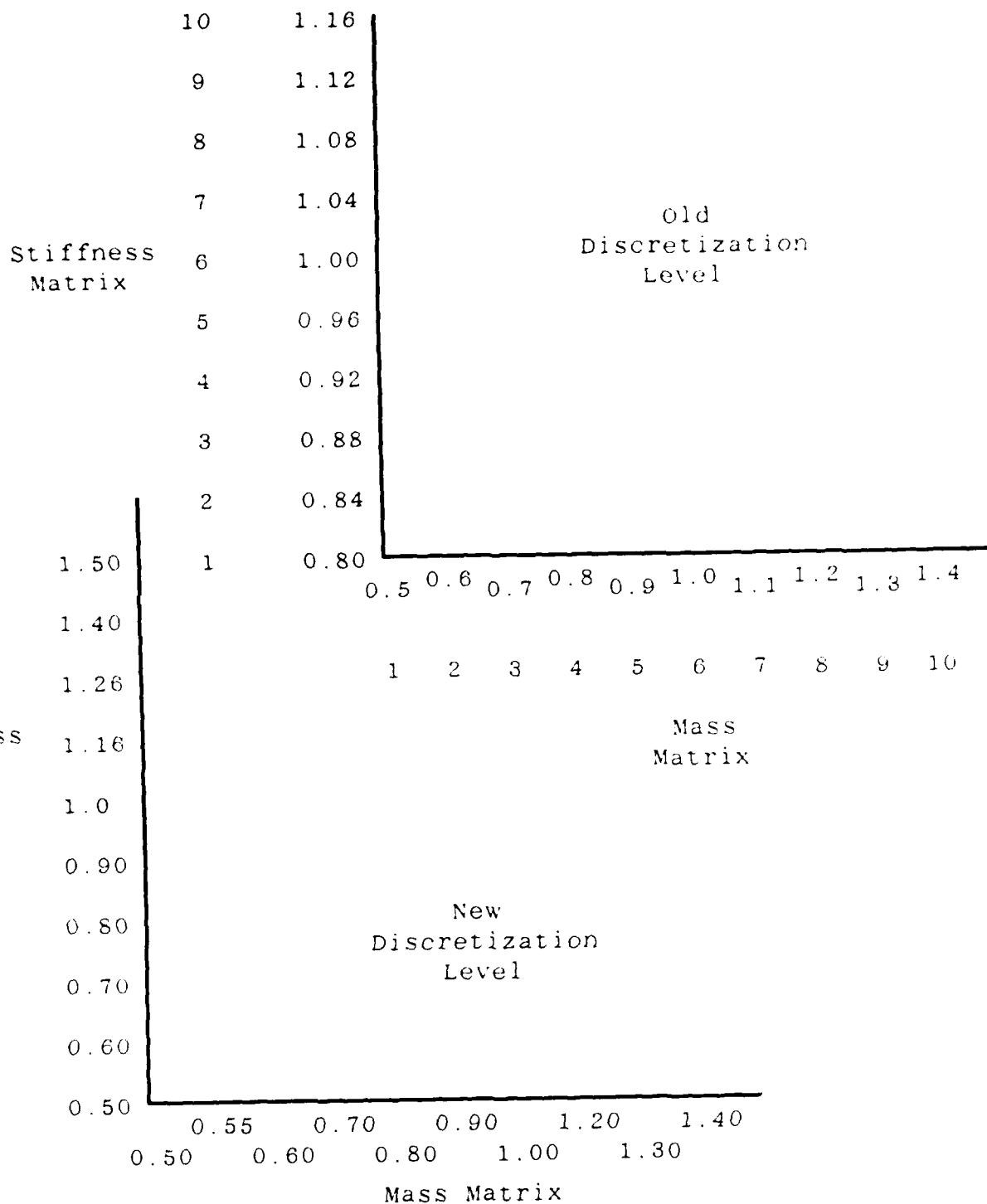


Fig. 5-11. Old and New Space Discretization Levels
Parameter #

given in Figure 5-11. Also note, that both the inconsistent spacing and the enlarged range of the stiffness and mass values were chosen to enhance parameter identifiability. After the optimal space discretization level was chosen, Figure 5-12 through Figure 5-15 were produced to compare with Figure 5-4 through Figure 5-7. Again these figures were produced using the same technique as the figures in the two-state model study in Section 5.3 except this model contains six states. The benefits of the new space discretization level can be seen, especially in Figure 5-12 when compared to Figure 5-4. In Figure 5-12, the moving-bank estimate of the true parameter locked onto the actual true parameter, while in Figure 5-4, the moving-bank estimate of the true parameter did not lock onto the actual true parameter. In addition, plots containing the state estimation errors were produced for the true parameters (2,7), (9,9), and (7,6) for the new space discretization level (see Figures 5-16 through 5-18). In comparing Figure 5-8 with Figure 5-16, the moving-bank algorithm estimate of the true parameter locked onto the actual true parameter (see Figure 5-12) and the error in the state estimate indicates the system is stable (see Figure 5-16), while in Figure 5-8 the moving-bank algorithm estimate of the true parameter did not lock onto the actual true parameter and the error in state estimate indicates that the system is unstable.

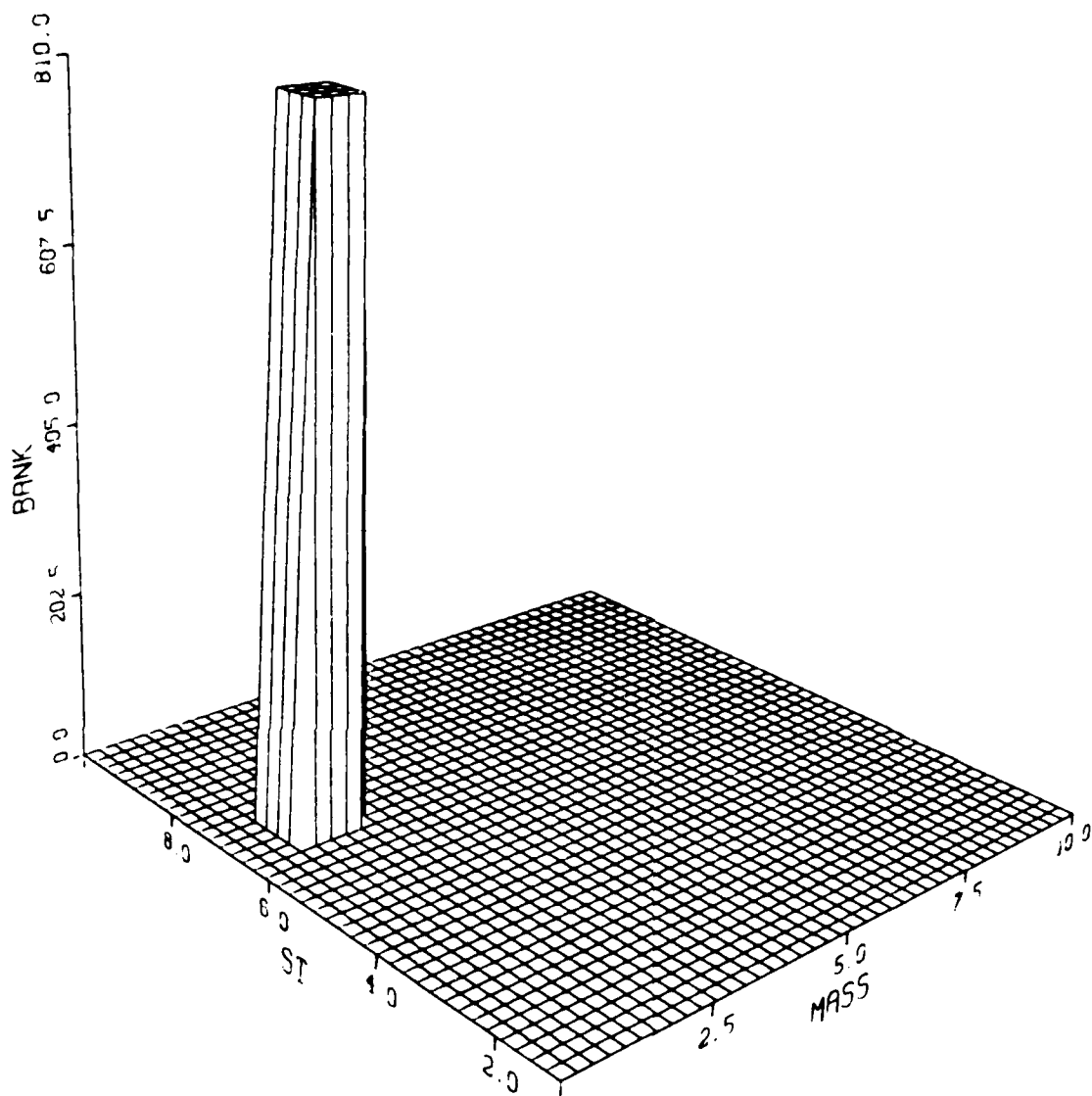


Fig. 5-12. Six-State Model Bank Location Time History;
 True Parameter at Mass = 2, Stiffness = 7
 New Discretization Level

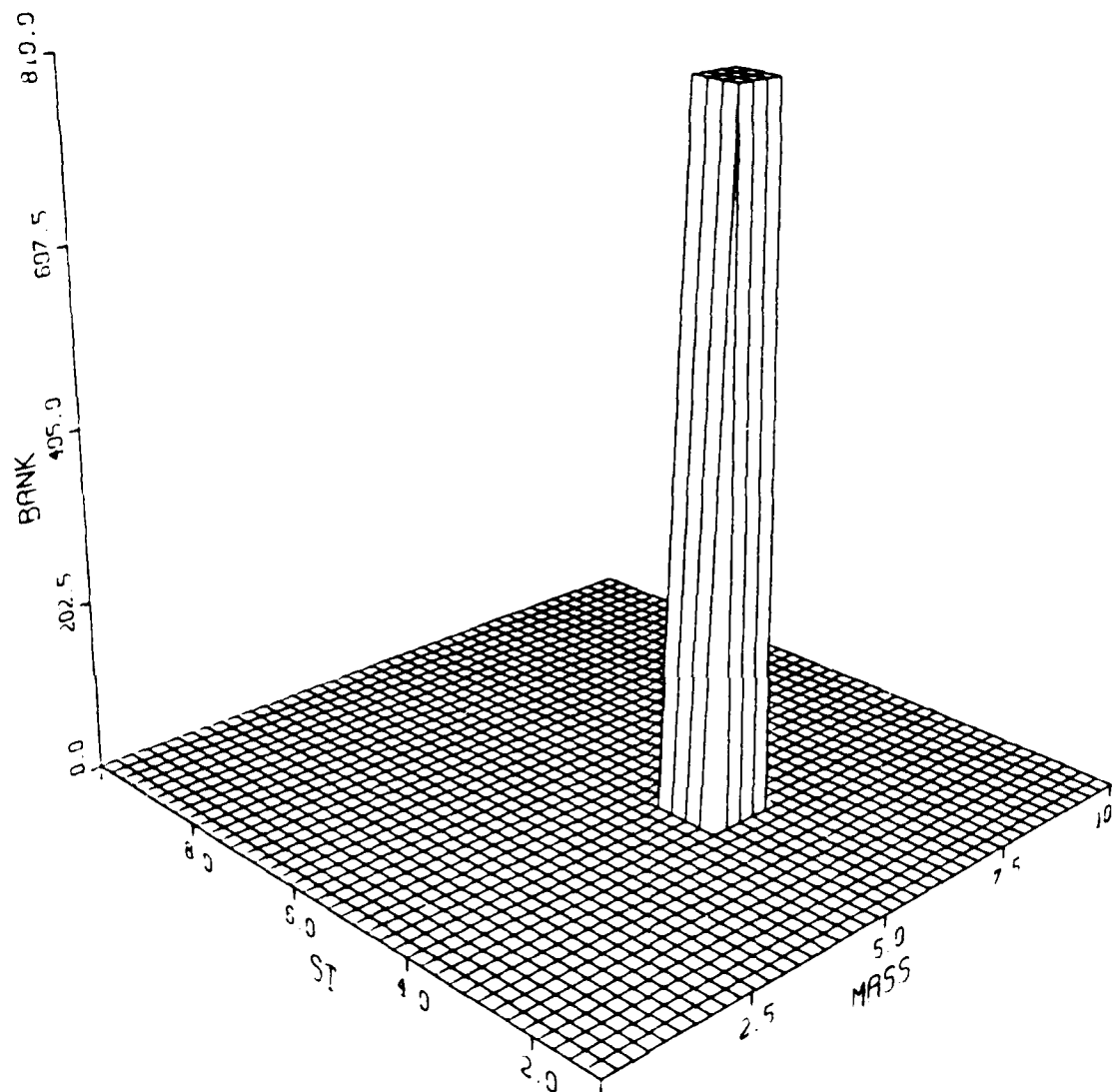


Fig. 5-13. Six-State Model Bank Location Time History;
True Parameter at Mass = 6, Stiffness = 4
New Discretization Level

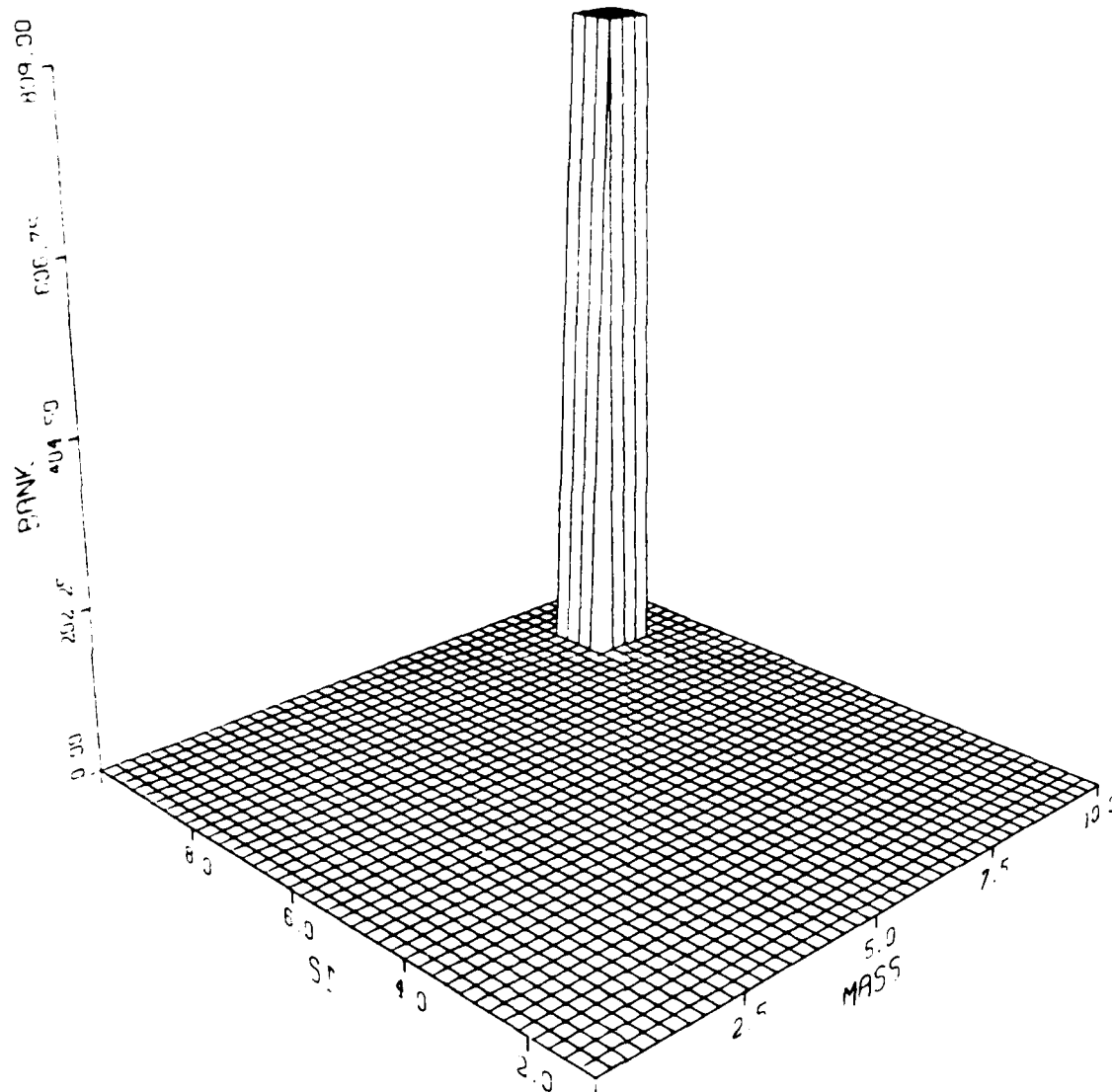


Fig. 5-14. Six-State Model Bank Location Time History;
True Parameter at Mass = 9, Stiffness = 9
New Discretization Level

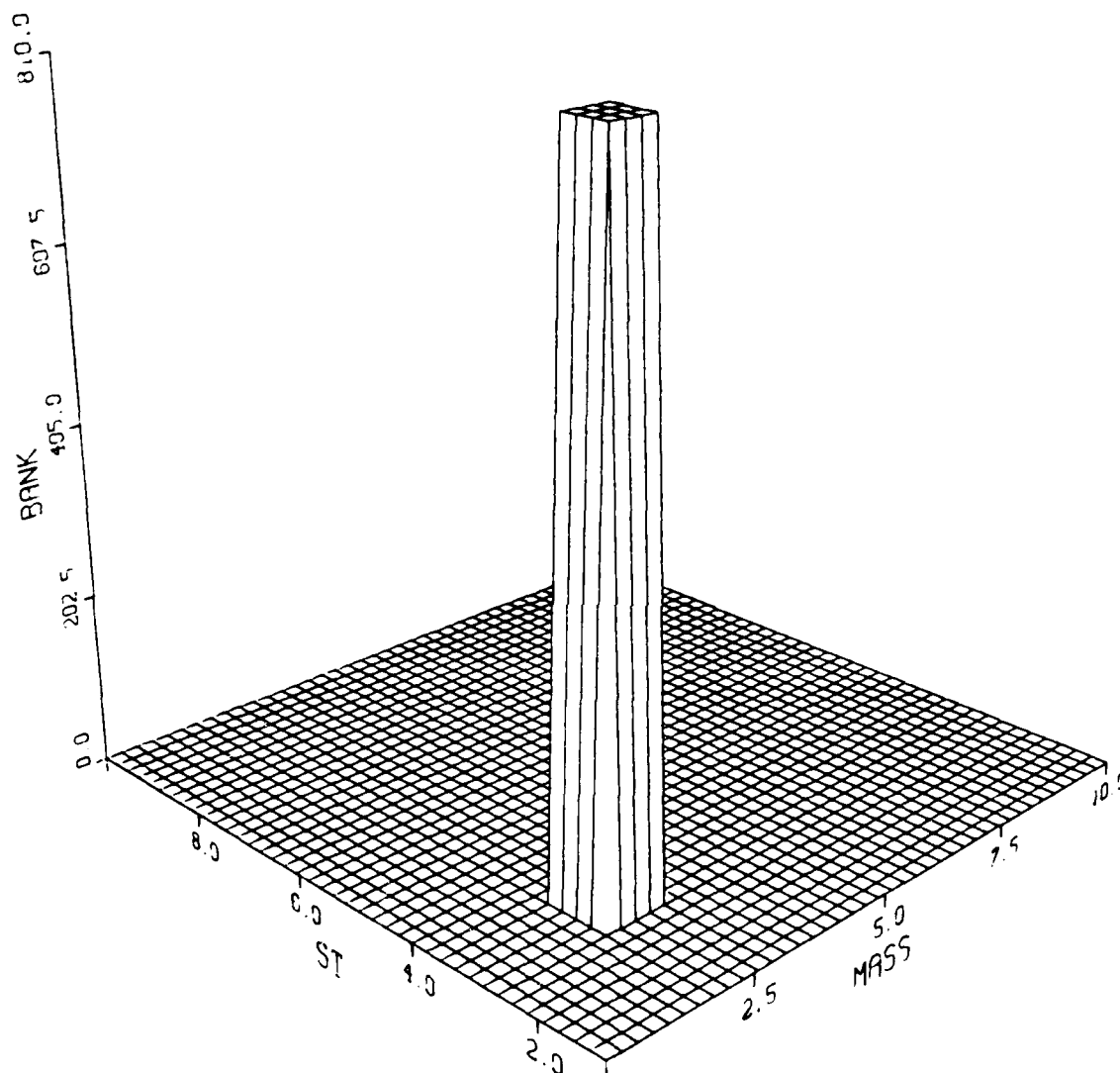


Fig. 5-15. Six-State Model Bank Location Time History;
True Parameter at Mass = 3, Stiffness = 3
New Discretization Level

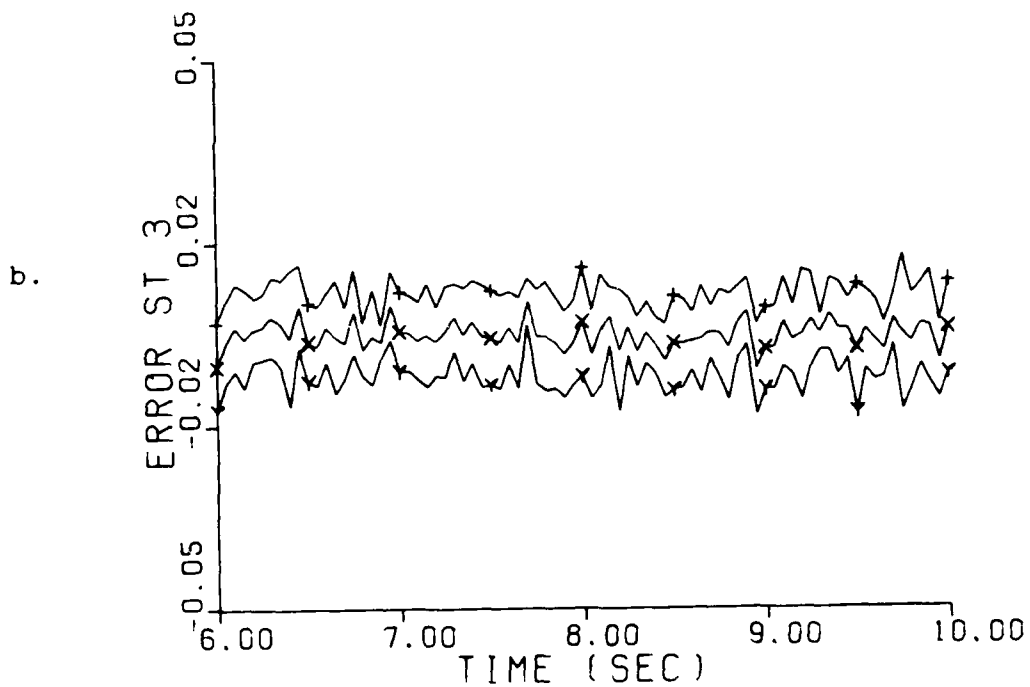
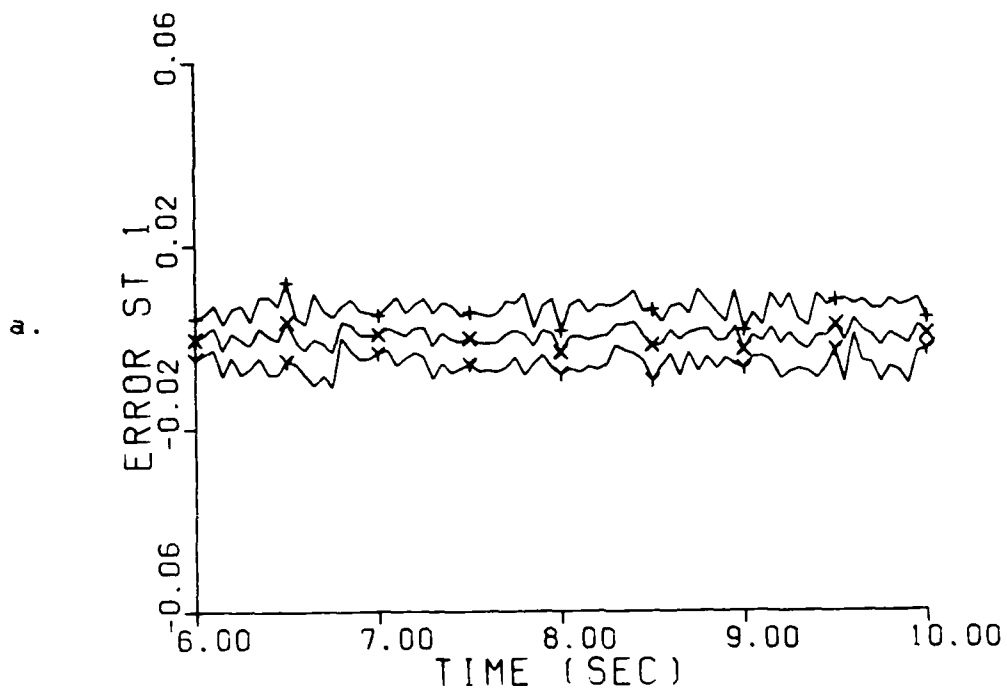


Fig. 5-16. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 New Discretization Level
 a. Position Error at Actuator #1 (in)
 b. Velocity Error at Actuator #1 (in/sec)

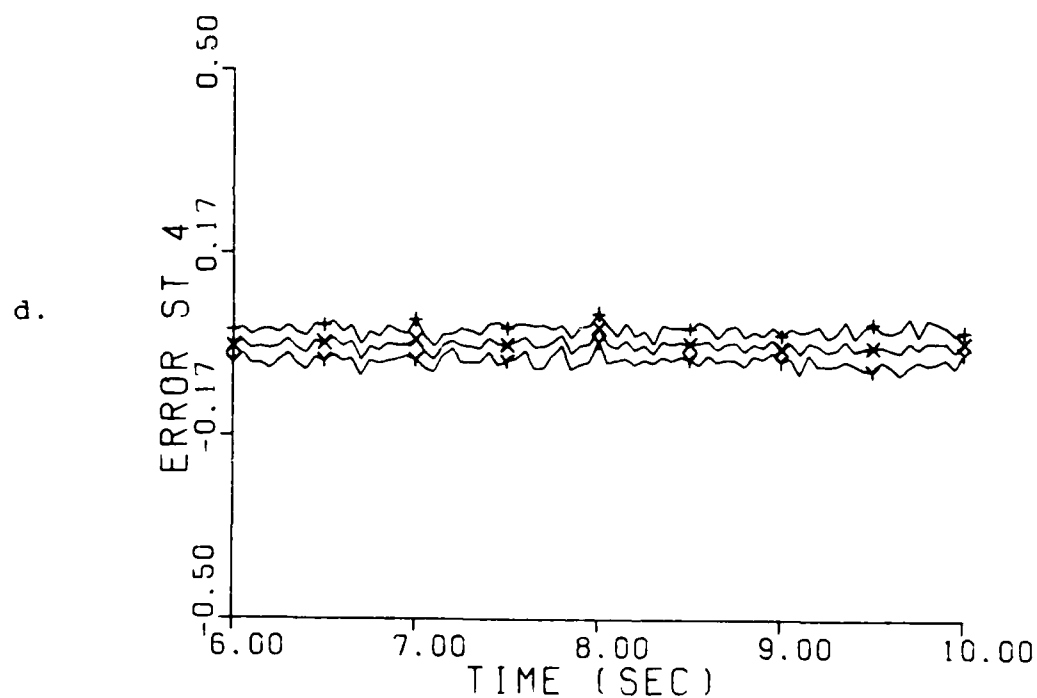
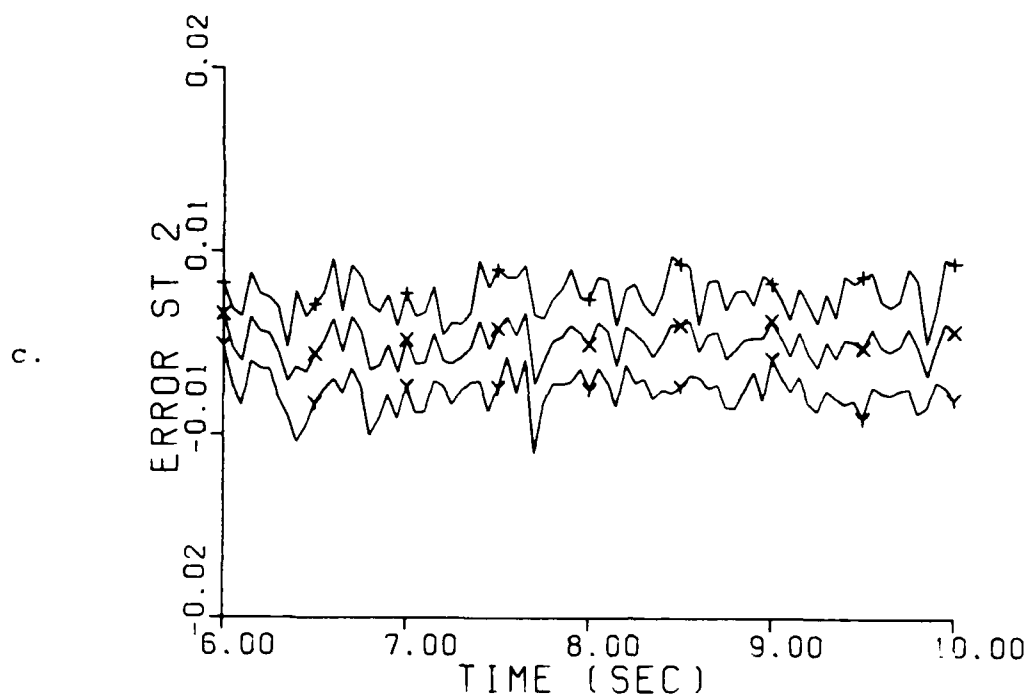


Fig. 5-16. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 New Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

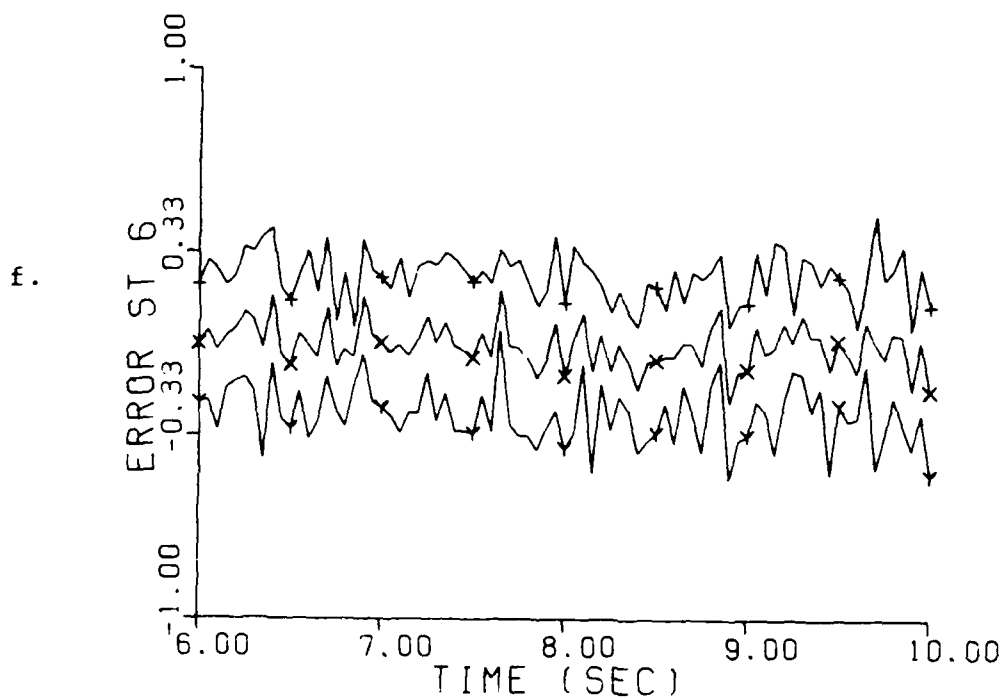
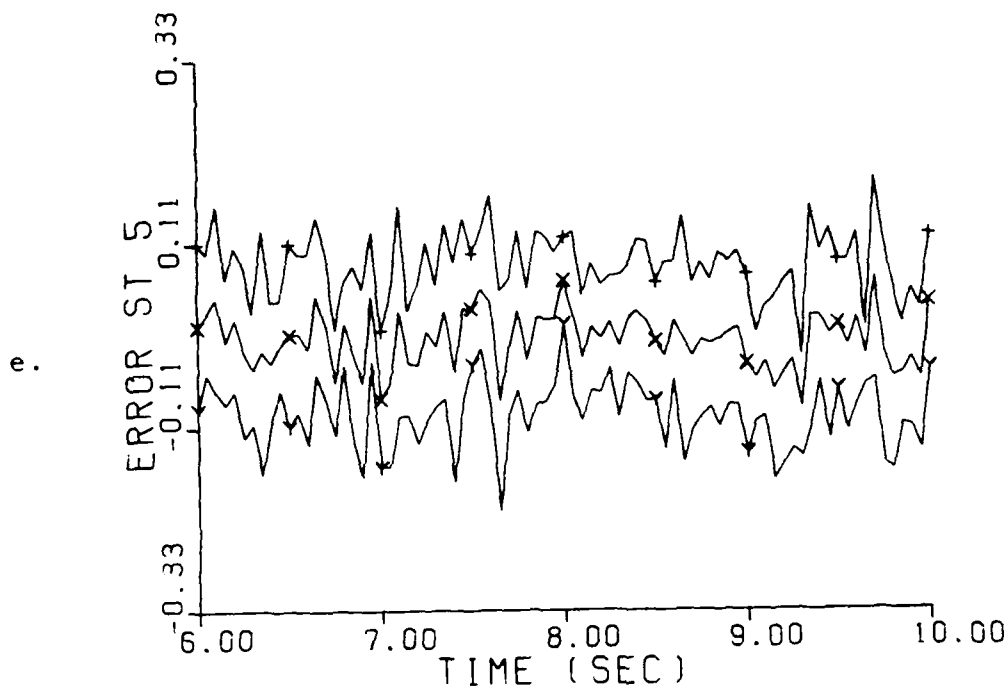


Fig. 5-16. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 2, Stiffness = 7
 New Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

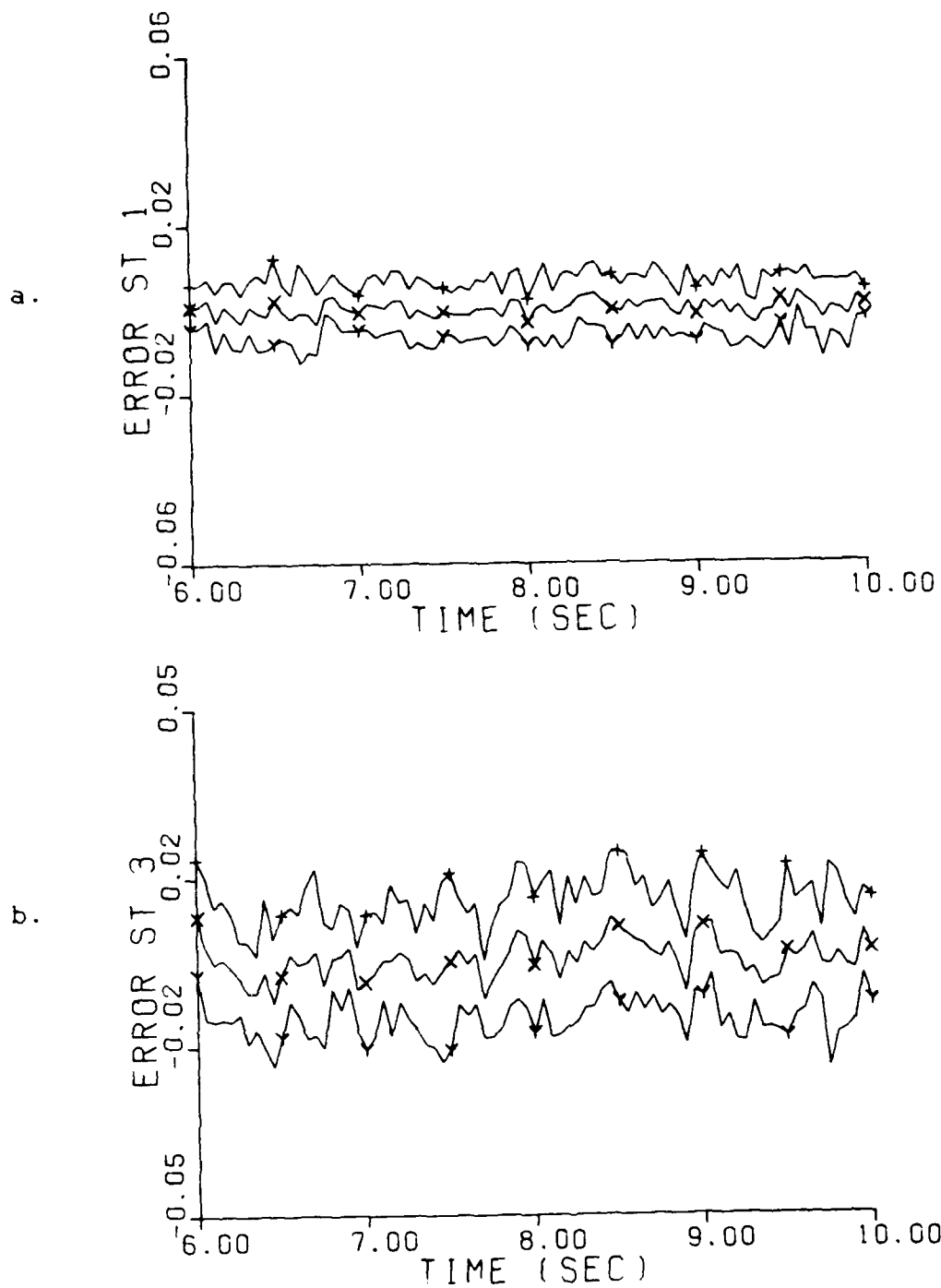


Fig. 5-17. State Estimation Errors: Mean ± 1 Std. Dev.
True Parameter at Mass = 9, Stiffness = 9
New Discretization Level
a. Position Error at Actuator #1 (in)
b. Velocity Error at Actuator #1 (in/sec)

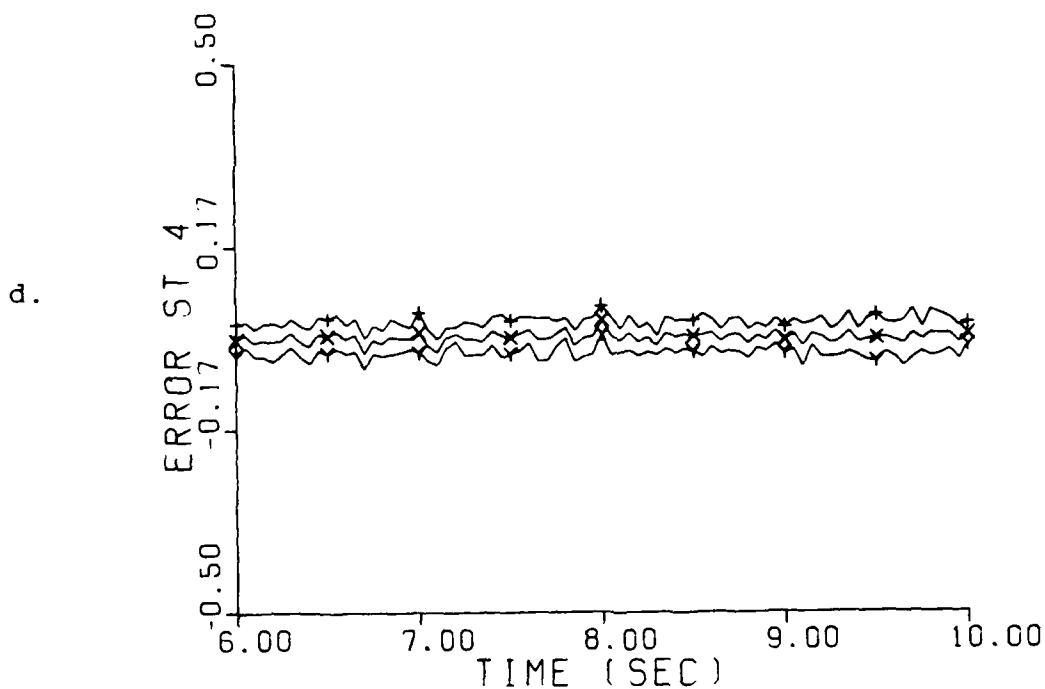
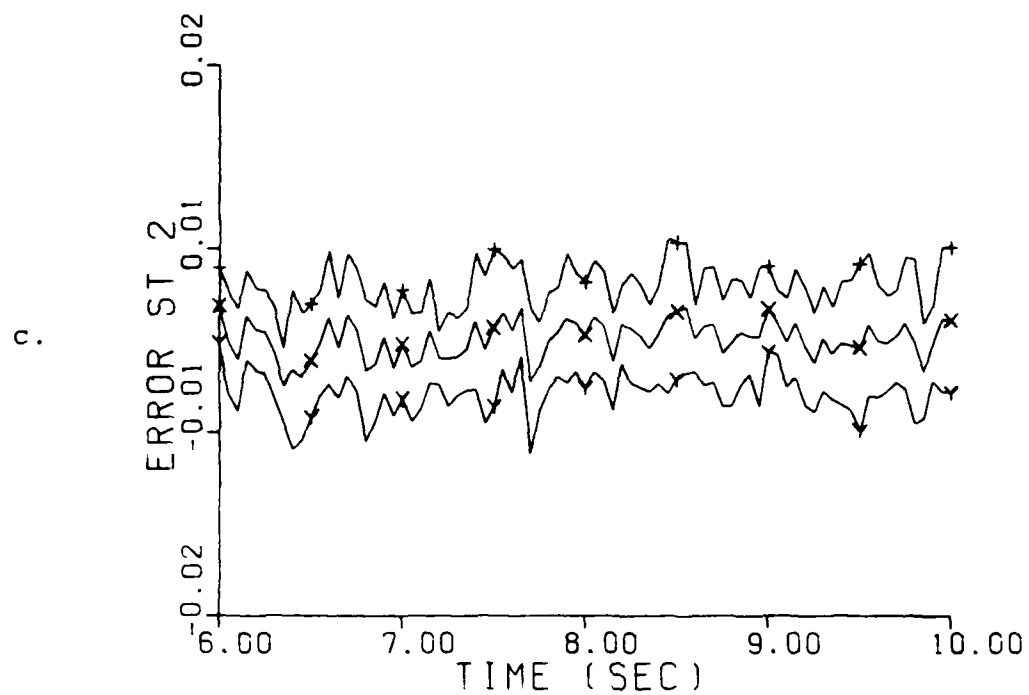


Fig. 5-17. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 9, Stiffness = 9
 New Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

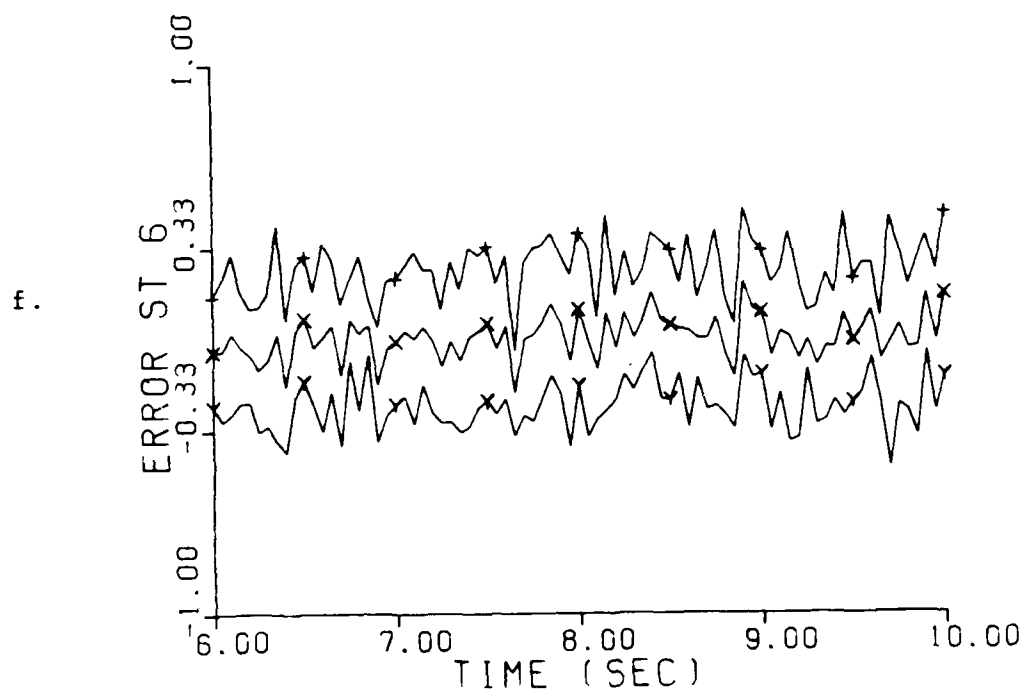
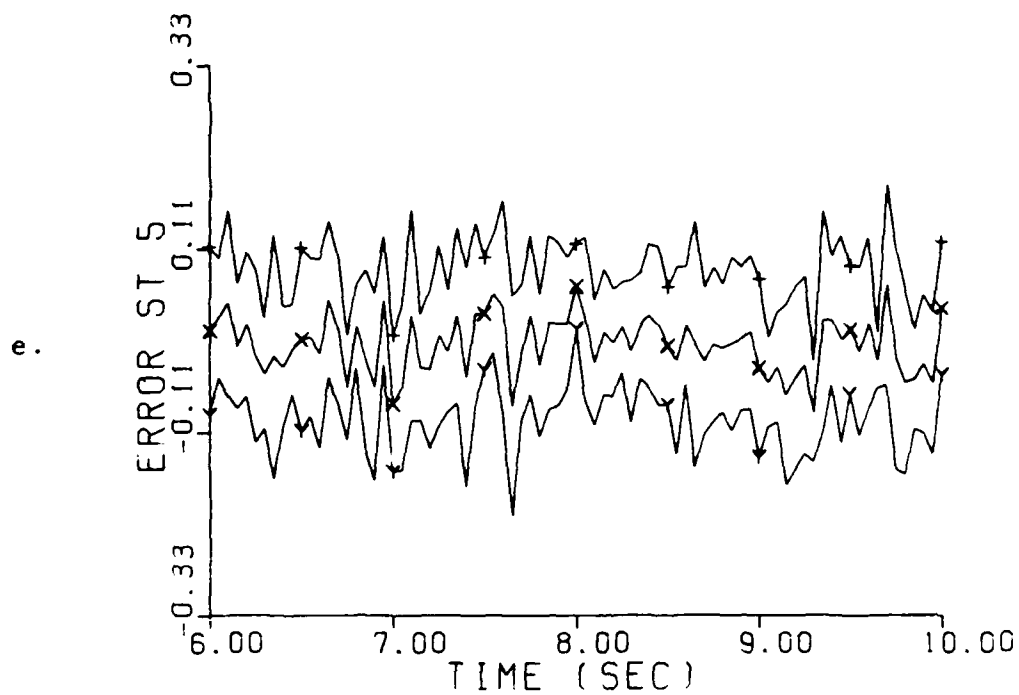


Fig. 5-17. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 9, Stiffness = 9
 New Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

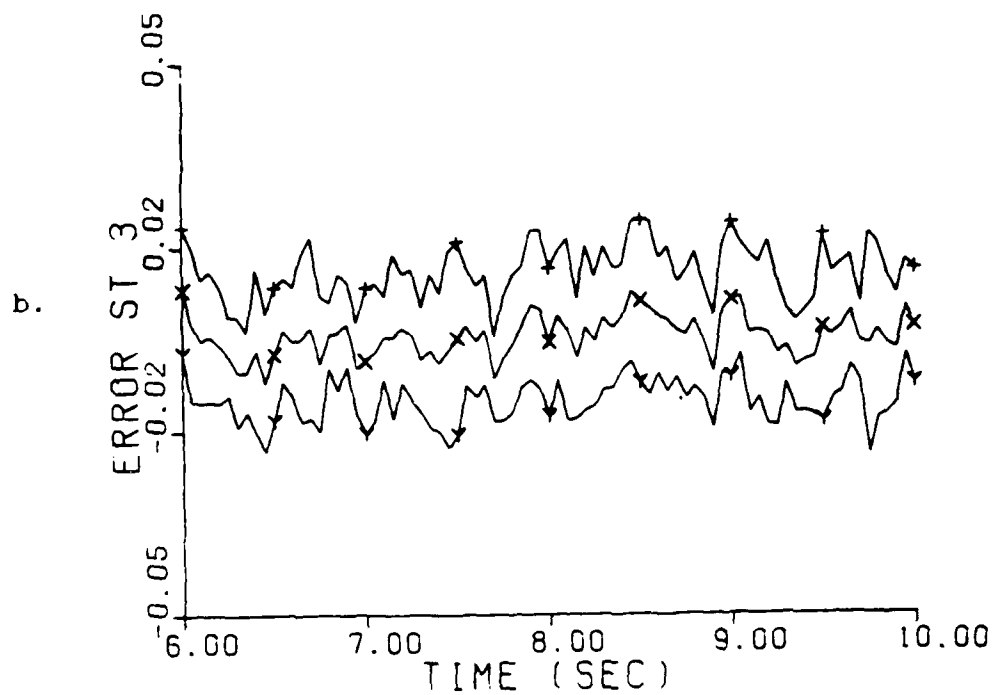
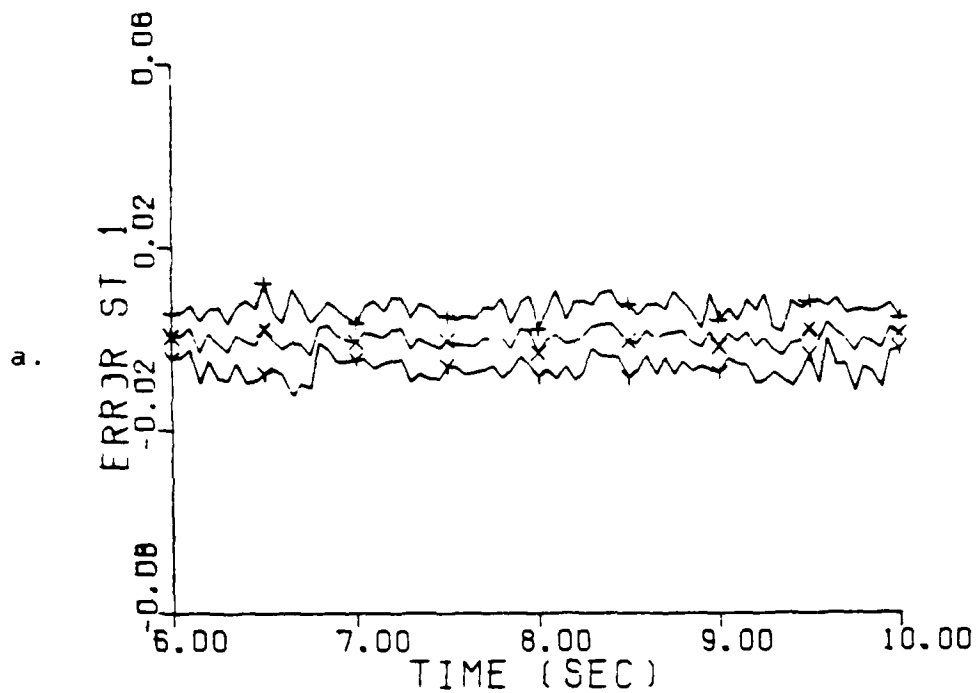


Fig. 5-18. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 New Discretization Level
 a. Position Error at Actuator #1 (in)
 b. Velocity Error at Actuator #1 (in/sec)

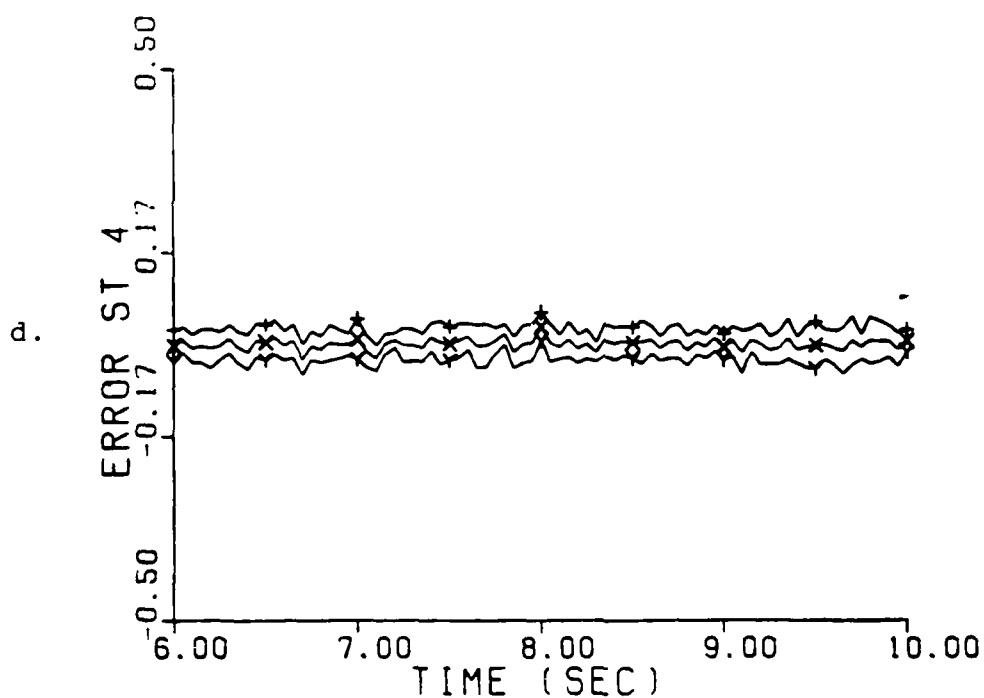
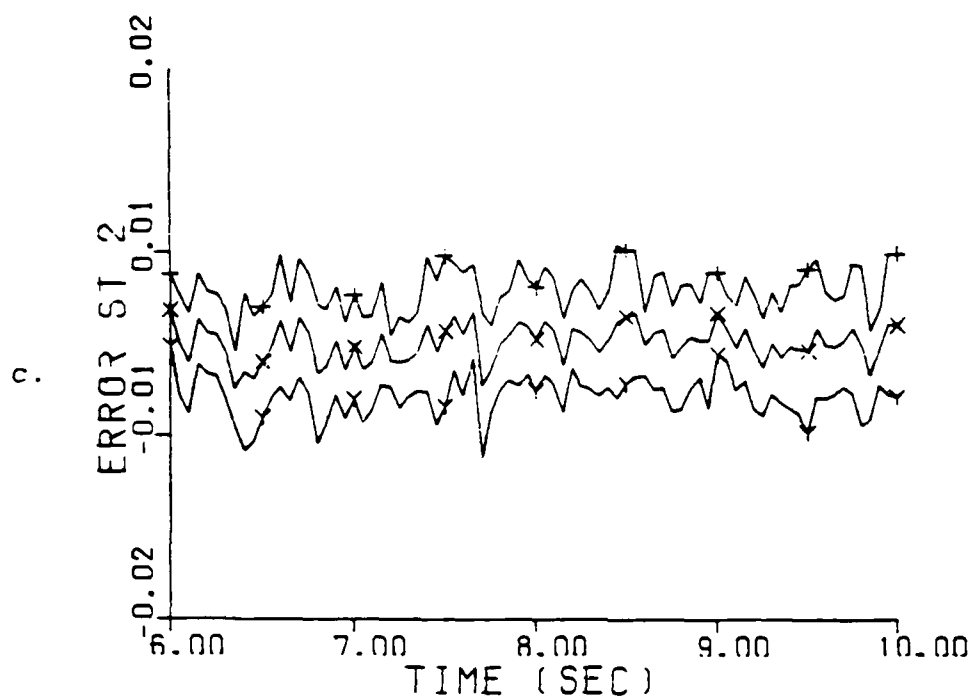


Fig. 5-18. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 New Discretization Level
 c. Position Error at Actuator #2 (in)
 d. Velocity Error at Actuator #2 (in/sec)

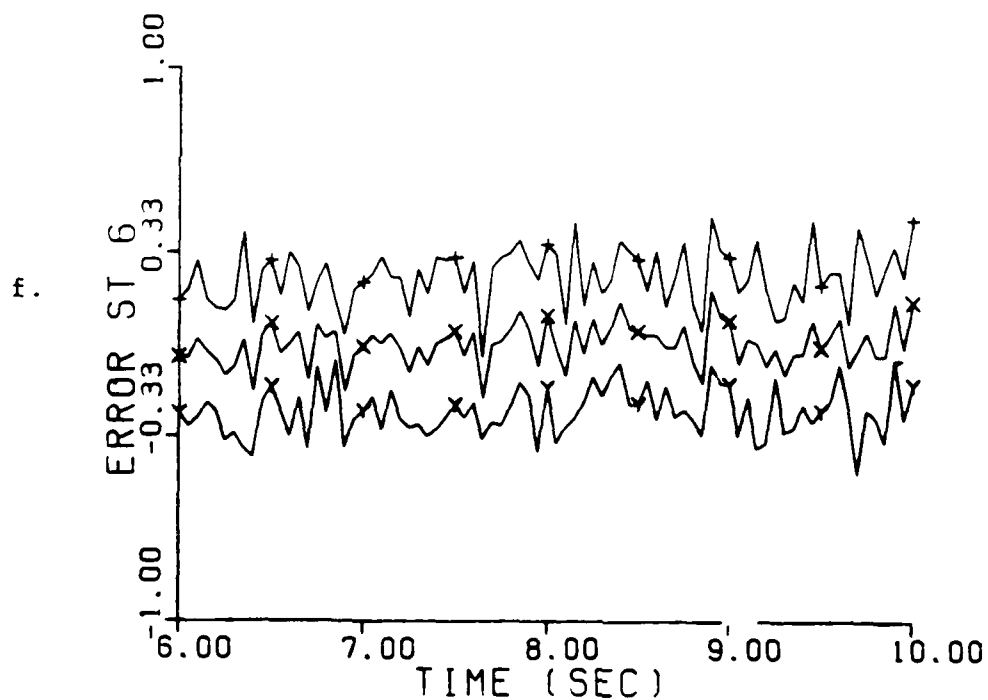
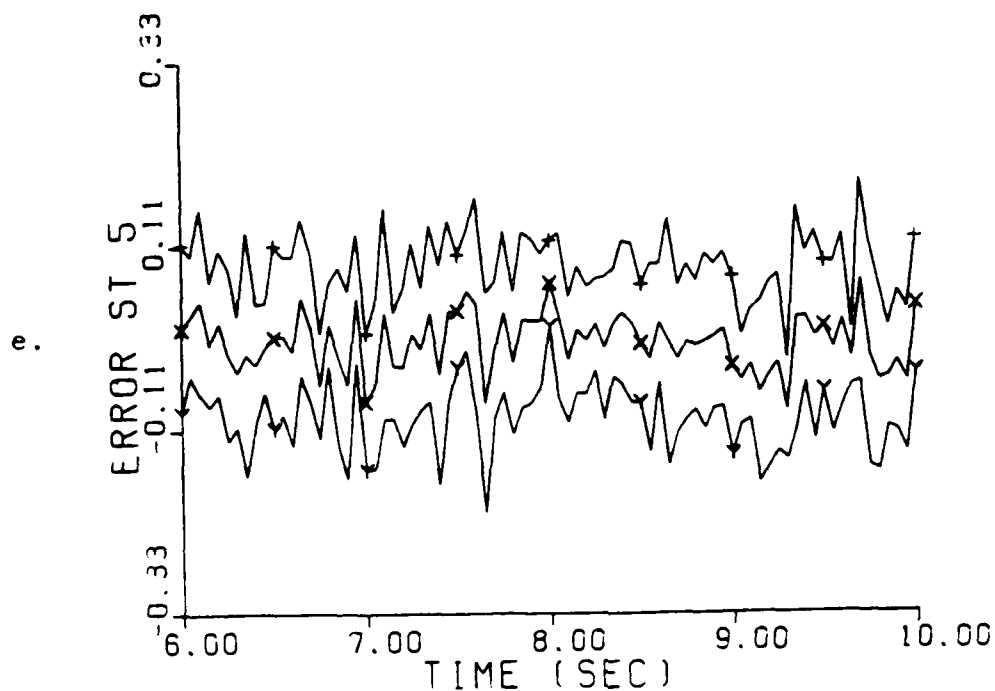


Fig. 5-18. State Estimation Errors: Mean ± 1 Std. Dev.
 True Parameter at Mass = 7, Stiffness = 6
 New Discretization Level
 e. Position Error at Actuator #3 (in)
 f. Velocity Error at Actuator #3 (in/sec)

5.6. Measurement and Dynamics Noise Sensitivity

Since the choice of the measurement noise covariance (R) and dynamics noise strength (Q) had such a dramatic effect on the performance of the moving bank in parameter estimation (i.e., if the Q and R values were chosen too large the moving-bank algorithm fails to operate correctly), a study was performed to determine when the moving-bank adaptive controller performance degraded to an unacceptable level as a function of the measurement noise covariance and dynamics noise strengths. Recall from Section 4.4.5, that the study was conducted by setting the Q and R values for the truth and filter models equal and then increasing these Q and R values one at a time. The results of this study are contained in Table 5-4. For some values of Q and R, the moving-bank algorithm showed no loss of adaptive performance even after the values were increased by eight orders of magnitude. These values are listed in Table 5-4 as showing no loss of adaptive performance. As is shown in the table, only three values significantly affected the performance of the moving bank in parameter, and these values are all contained in the measurement covariance matrix. However, it makes sense that the moving bank would be more sensitive to R values because noisy measurement values would tend to mask the difference between good and bad assumed models within

Table 5-4

Dynamics Noise Strength and Measurement Noise Covariance

Maximum Values for Acceptable Moving-Bank Performance

Dynamics Noise Strength

(Units: Position - in^2/sec ; Velocity - in^2/sec^3)

Noise Input at node 1 (Q1) - No loss of adaptive performance

Noise Input at node 2 (Q2) - No loss of adaptive performance

Noise Input at Hub (Q3) - No loss of adaptive performance
at angular velocity level

Noise Input at Hub (Q4) - No loss of adaptive performance
at angular acceleration level

Measurement Covariance Noise

(Units: Position - in^2/sec ; Velocity - in^2/sec)

Position Sensor at Node 1 (R1) - 0.27

Position Sensor at Node 2 (R2) - 0.027

Velocity Sensor at Node 1 (R3) - No loss of adaptive performance

Velocity Sensor at node 2 (R4) - 4.2668

Position Sensor at Hub (R5) - No loss of adaptive performance

Velocity sensor at Hub (R6) - No loss of adaptive performance

elemental filters, making it hard to discern the effect of the system parameters. Figure 5-19 through Figure 5-22 were produced using the same technique as in the figures in Section 5.3 with R1, R2, and R4 simultaneously set to the values given in Table 5-4. These figures show that the estimate of the true parameter was beginning to degrade, as compared to the performance seen in Figures 5-12 through 5-15. For example, in Figure 5-19 the moving-bank estimate of the true parameter does not lock onto the actual true parameter, while in Figure 5-12 the estimate of the true parameter does lock onto the actual true parameter. To show that this degradation continues to the point where the moving bank completely fails to operate, Figure 5-23 and Figure 5-24 were generated. Again these figures were generated using the same techniques as in figures in Section 5.3. In both Figure 5-23 and Figure 5-24, R1, R2, and R4 were increased an additional 3 orders of magnitude beyond the values in Table 5-4. Both figures show that the moving bank does not move off the (5,5) parameter point, which is where the moving bank is initially centered. Therefore, there were upper limits on the R values, for viable parameter estimation. These upper limits are important because they determine the precision level the sensors must meet for the moving-bank algorithm to operate correctly.

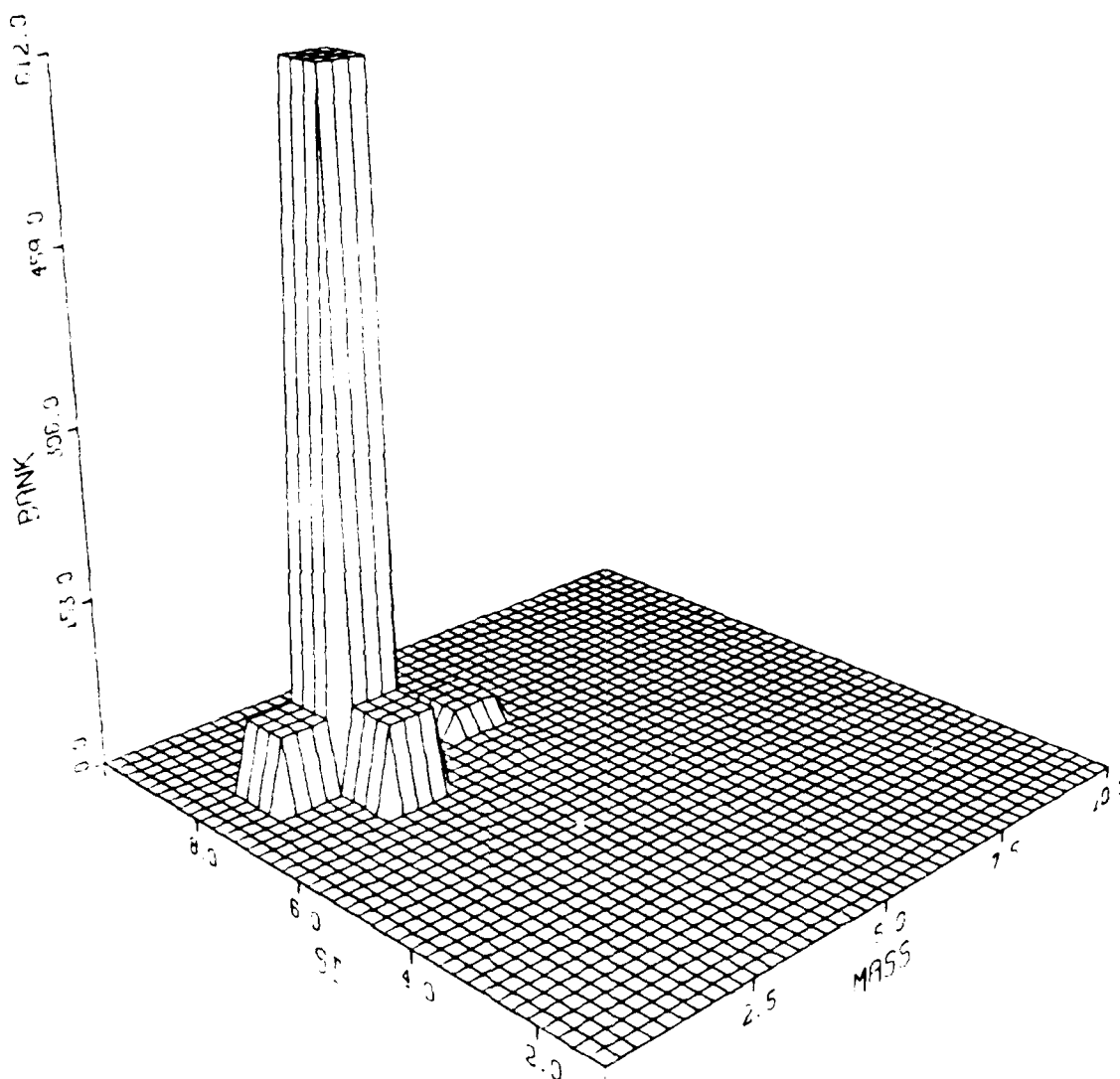


Fig. 5-19. Six-State Model Bank Location Time History;
True Parameter at Mass = 2, Stiffness = 7
R1, R2, and R4 Set to Maximum Value

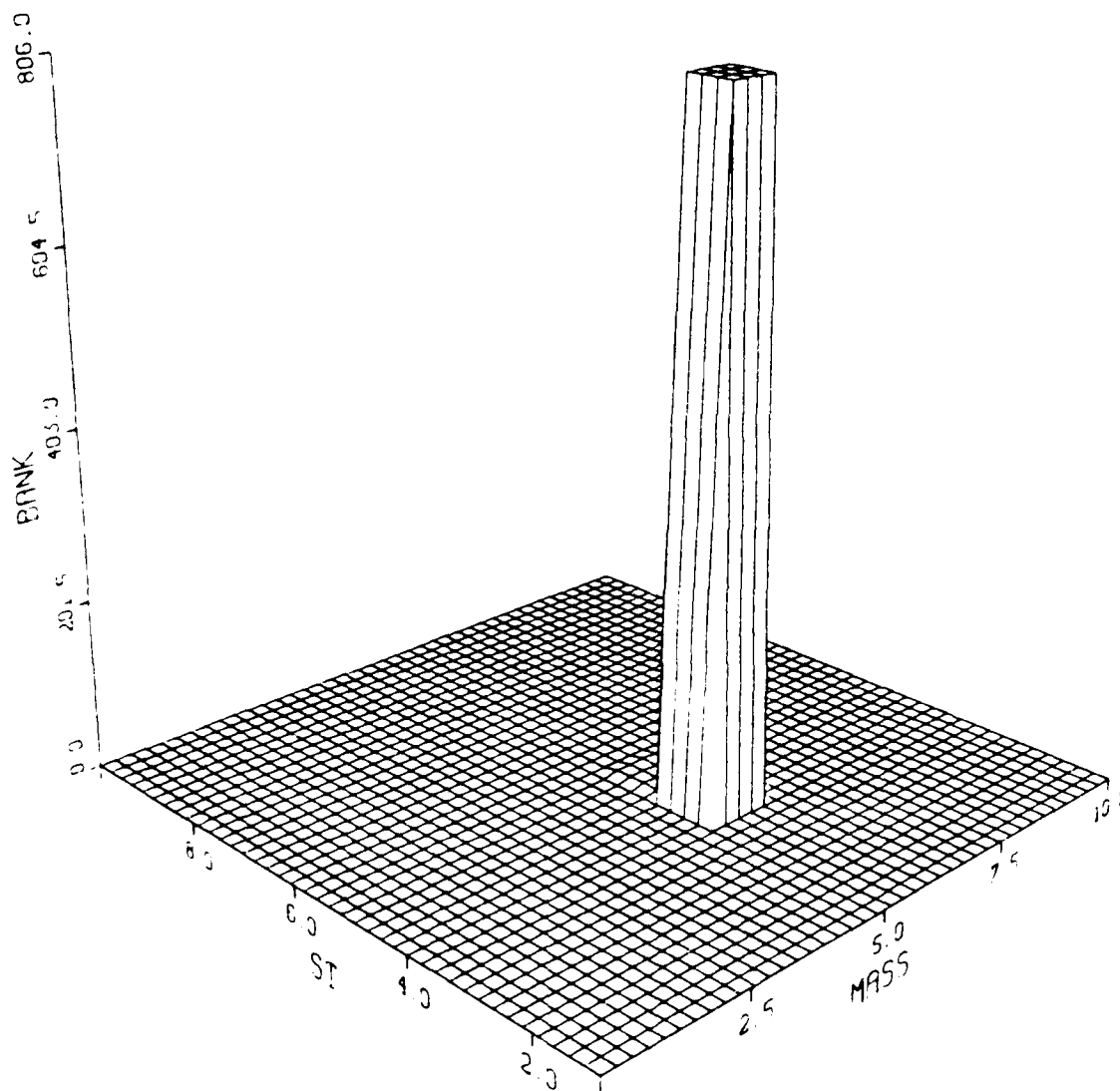


Fig. 5-20. Six-State Model Bank Location Time History;
True Parameter at Mass = 6, Stiffness = 4
R1,R2, and R4 Set to Maximum Value

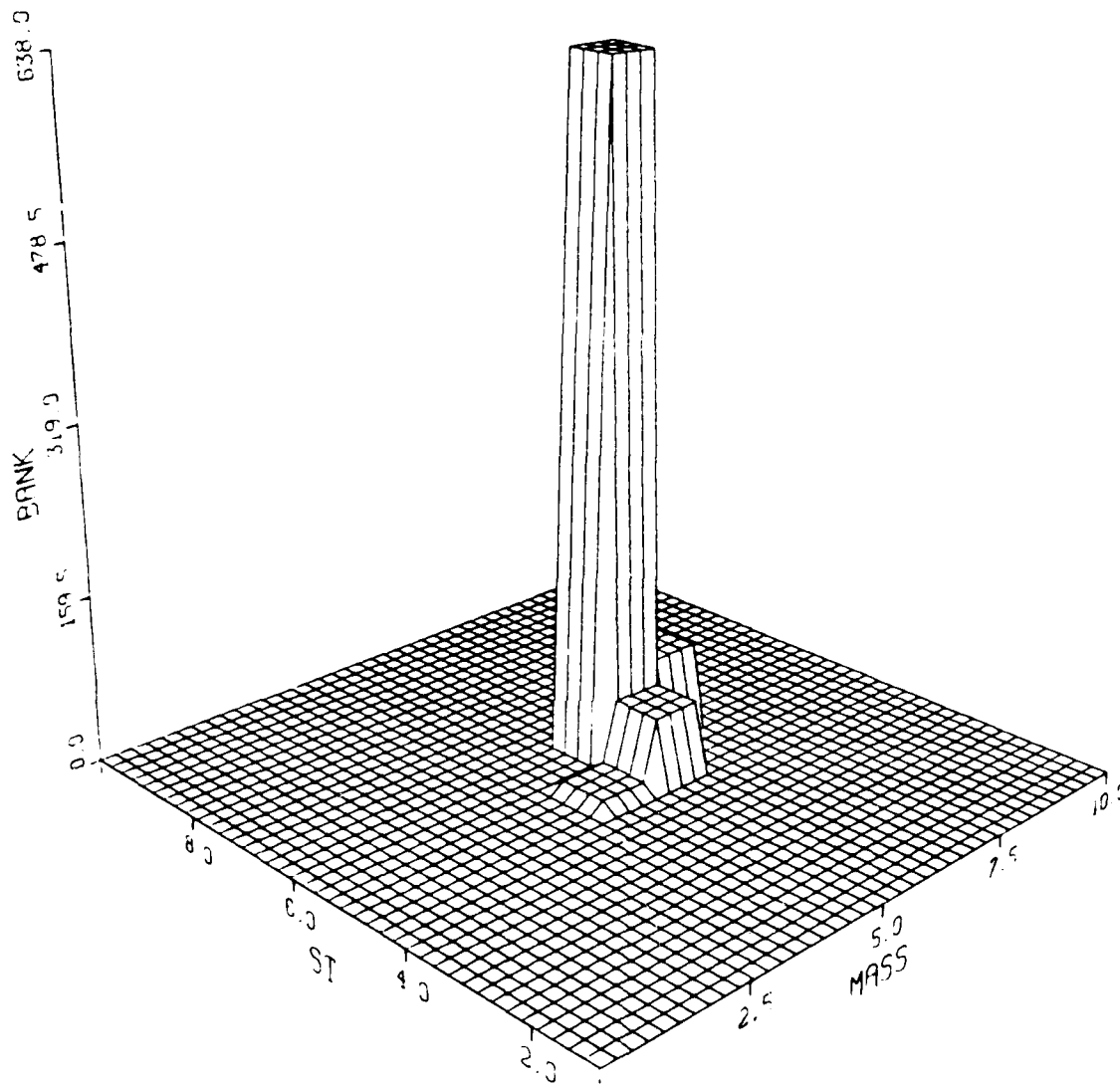


Fig. 5-21. Six-State Model Bank Location Time History;
 True Parameter at Mass = 9, Stiffness = 9
 R1,R2, and R4 Set to Maximum Value

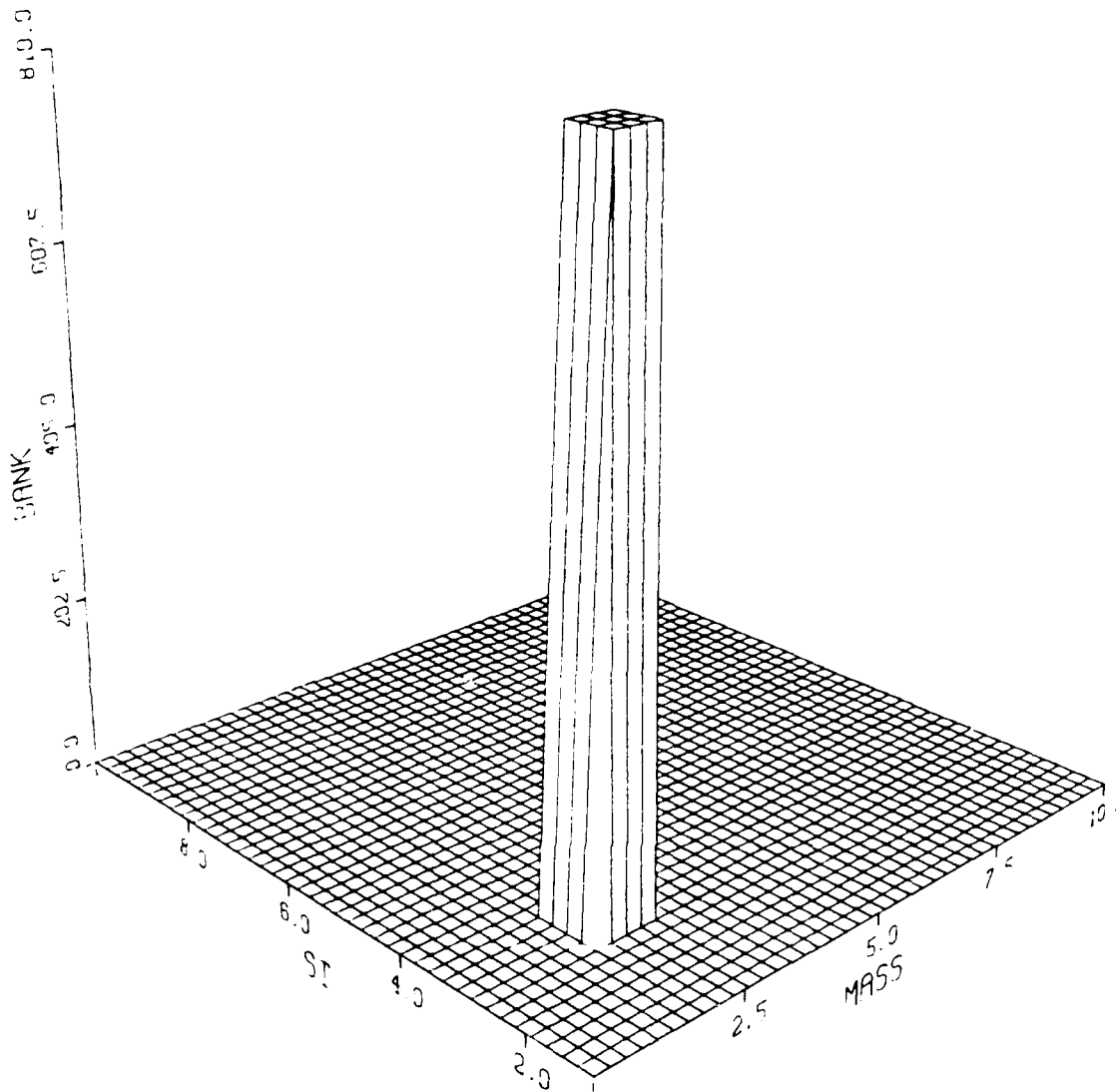


Fig. 5-22. Six-State Model Bank Location Time History;
True Parameter at Mass = 3, Stiffness = 3
R1, R2, and R4 Set to Maximum Value

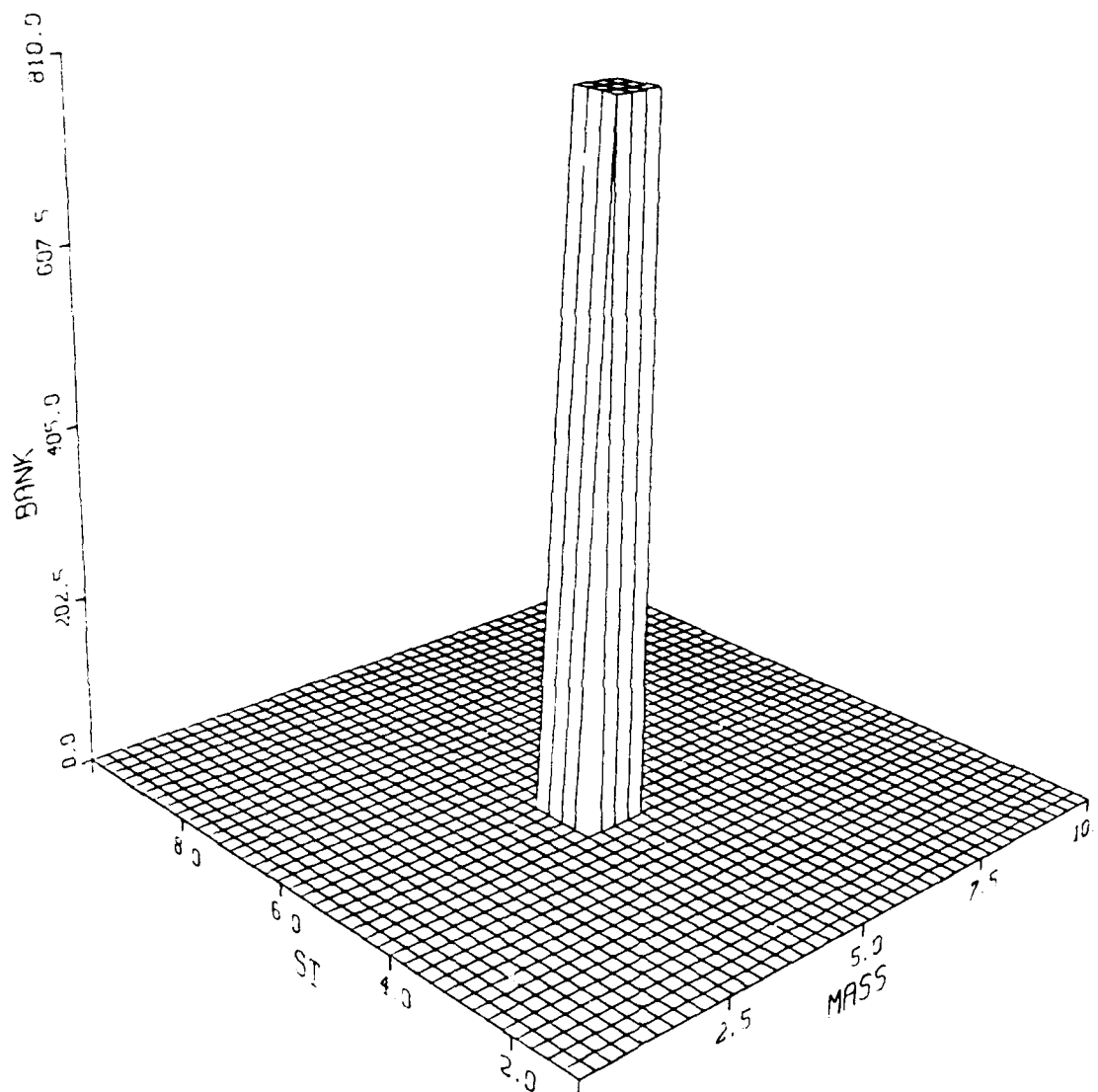


Fig. 5-23. Six-State Model Bank Location Time History;
True Parameter at Mass = 1, Stiffness = 1
R1, R2, and R4: Three Orders of Magnitude
Larger Than In Table 5-2

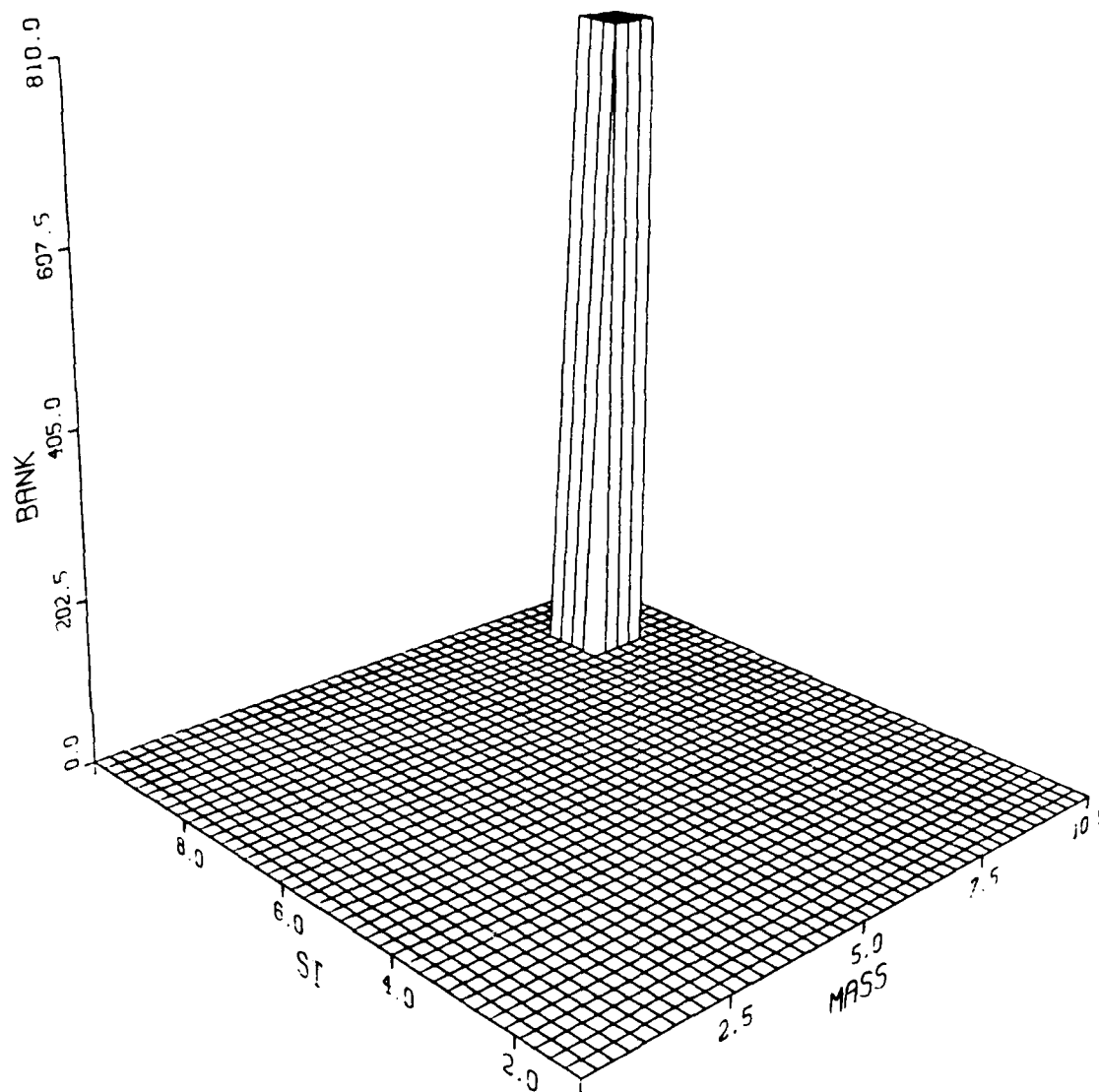


Fig. 5-24. Six-State Model Bank Location Time History;
 True Parameter at Mass = 9, Stiffness = 9
 R1, R2, and R4: Three Orders of Magnitude
 Larger Than In Table 5-2

5.7. Controller Design: Need for Cross Terms in the Quadratic Cost

After the MMAE began to supply accurate state estimation, a study was conducted to determine if additional performance could be attained by adding cross terms to the quadratic cost function. The generalized cost function is given in Equation (5-1) [23:73]. The LQ controller is developed fully in Appendix A.

$$J = E \left(\frac{1}{2} \underline{x}^T(t_{N+1}) X_f \underline{x}(t_{N+1}) + \sum_{i=1}^N \left(\frac{1}{2} [\underline{x}^T(t_i) X(t_i) \underline{x}(t_i) + \underline{u}^T(t_i) U(t_i) \underline{u}(t_i) + 2 \underline{x}^T(t_i) S(t_i) \underline{u}(t_i)] \right) \right) \quad (5-1)$$

where

$\underline{x}(t_i)$ = state vector

$\underline{u}(t_i)$ = control input vector

$X(t_i)$ = state weighting matrix

$U(t_i)$ = control weighting matrix

$S(t_i)$ = cross term matrix

The cross term arises due to the fact that control is being applied over the entire sample period, not just at the discrete sample times [23:75]. Recall that the differential equations given in Section 4.4.6 [22:76] were solved to determine the state weighting matrix (X), the control weighting matrix (U), and the cross term weighting

matrix (S). In addition because the matrices were of different dimensions, SS^T was formed and then the square root of the eigenvalues of the SS^T matrix was used to provide a comparison with the eigenvalues of the X matrix. The eigenvalues for the U, X, and, SS^T matrices are given in Table 5-5. This table shows that the eigenvalues of the SS^T matrix are at least an order of magnitude smaller than the eigenvalues of the X matrix. Therefore, the cross weighting matrix does not attribute greatly to the cost function and was not added to the quadratic cost function for the remainder of the research. However, if the sample period were chosen smaller, then the cross weighting matrix would have a larger effect on the cost function. The corresponding state control performance will be seen in Section 5.9.

5.8. State Weighting and Control Weighting Matrix Tuning

After the decision was made not to include a cross weighting matrix, a study was performed to tune the state weighting matrix (X) and the control weighting matrix (U). Recall from Section 4.4.7, that the tuning was performed for the (7,6) parameter point. The criteria used for tuning was to choose the control weighting matrix (U) small (diagonal, with terms of 3, 3, and 9 along the diagonal) and increase the state weighting matrix until the rms error

Table 5-5

Comparison of Eigenvalues of State Weighting Matrix,
Control Weighting Matrix, and Cross Term Matrix

State Weighting Matrix X

X1 - 0.98168
X2 - 0.1615808
X3 - 0.630921
X4 - 0.001221
X5 - 0.0012396
X6 - 0.9891725

Control Weighting Matrix U

U1 - 1.70465
U2 - 0.0005257
U3 - 0.00063244

Cross Term Matrix SS^T

S1 - 0.0
S2 - 0.0
S3 - 0.0
S4 - 0.0001252
S5 - 0.0000998

on the true states stop changing drastically. The X and U matrices were tuned for the (7,6) parameter point, meaning that the filter located at parameter point (7,6) was matched against the truth model located at parameter point (7,6). The (7,6) parameter point was chosen so that the parameter point would be near the center of the space but not exactly corresponding to a fixed-bank parameter point of a fixed-bank algorithm with coarse discretization: the nine filters included in the fixed-bank were located at parameter points (1,1), (1,5), (1,9), (5,1), (5,5), (5,9), (9,1), (9,5), and (9,9). The final values of the X and U matrices are given in Table 5-6. The X values were chosen

Table 5-6

State Weighting and Control Weighting Matrices
 (for the (7,6) parameter point)
 (Units: position - in; velocity - in/sec)

Position of Node 1 (X1) - 762,700.0
 Position of Node 2 (X2) - 7917.8
 Velocity of Node 1 (X3) - 508.5
 Velocity of Node 2 (X4) - 82.0
 Position of Hub (X5) - 1.
 Velocity of Hub (X6) - 1.0
 Actuator at Node 1 (U1) - 3.0
 Actuator at Node 2 (U2) - 3.0
 Actuator at Hub (U3) - 9.0

intentionally large as compared to the U values, thereby causing a large emphasis to be placed on quelling the state estimates to zero while not strongly caring how much control energy it takes to accomplish this. The impact of tuning the X and U matrices will be addressed further in Section 5.9.

5.9. Controller Study

Once the X and U values were determined, a controller study was performed to compare the performance of the moving-bank controller with a fixed-bank controller and a benchmark controller. The controller used in this study is shown in Figure 5-25. In the operation of this adaptive algorithm, the parameter estimate locked onto the true parameter in less than 0.75 sec. This, coupled with the fact that a dither signal was being applied to the system for 0.5 sec, was the basis for not using a transition period of a partially adaptive controller as described in Section 2.4, and instead going straight into the controller scheme set up in Figure 5-25. Therefore, the remainder of this effort was performed with an open loop dither signal of magnitude 300 and frequency 30 rad/sec being applied for the first 0.5 sec. The magnitude, frequency, and duration of the dither signal was determined by trial and error. The dither signal was to excite the system but not to act

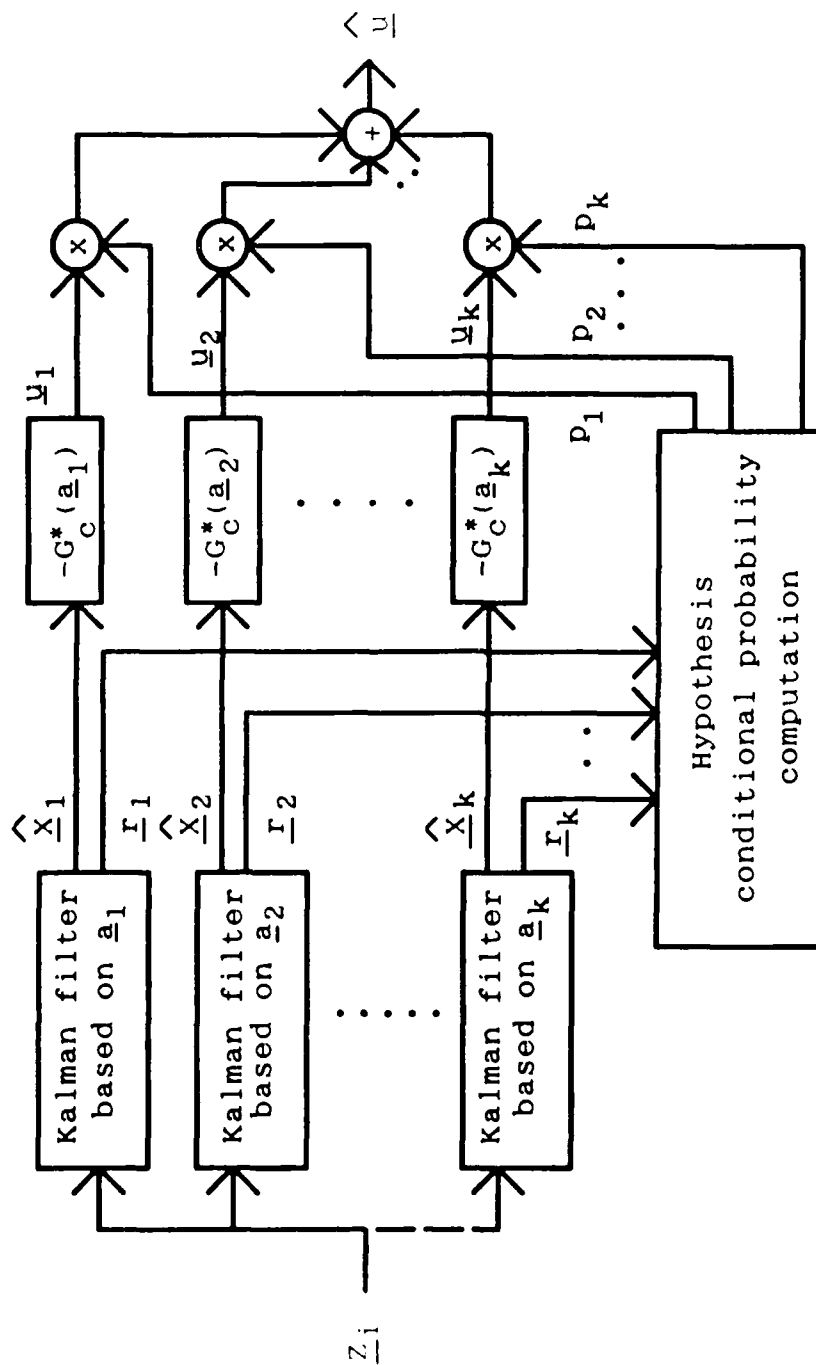


Fig. 5-25. Multiple Model Adaptive Controller

as the major forcing function. Following the period of applying the open loop dither signal, multiple model adaptive control was applied to the system. The complete details on the steps to accomplish this study are given in Section 4.4.8.

In this study, a benchmark controller was needed to compare with the moving-bank and fixed-bank multiple model adaptive controllers. This benchmark controller was established by using a single filter and controller located at parameter (7,6) matched against a truth model at parameter (7,6). The controller scheme used in the benchmark controller is given in Figure 4-1, except the benchmark controller does not estimate the true parameter (a). The benchmark controller performance for parameter (7,6) is given in Figure 5-26. This figure plots the mean value (symbol x), the mean plus 1 sigma (symbol +), and the mean minus 1 sigma (symbol y) of each of the six true states and three control inputs for the entire length of the runs. The purpose of this figure is to show that the transients are short lived and stable. Therefore, in the remainder of this section only the post-transient performance is given in the figures.

In order to compare the post-transient performance of the controllers effectively, Figures 5-27 to 5-29 were constructed exactly like Figure 5-26, except the plots

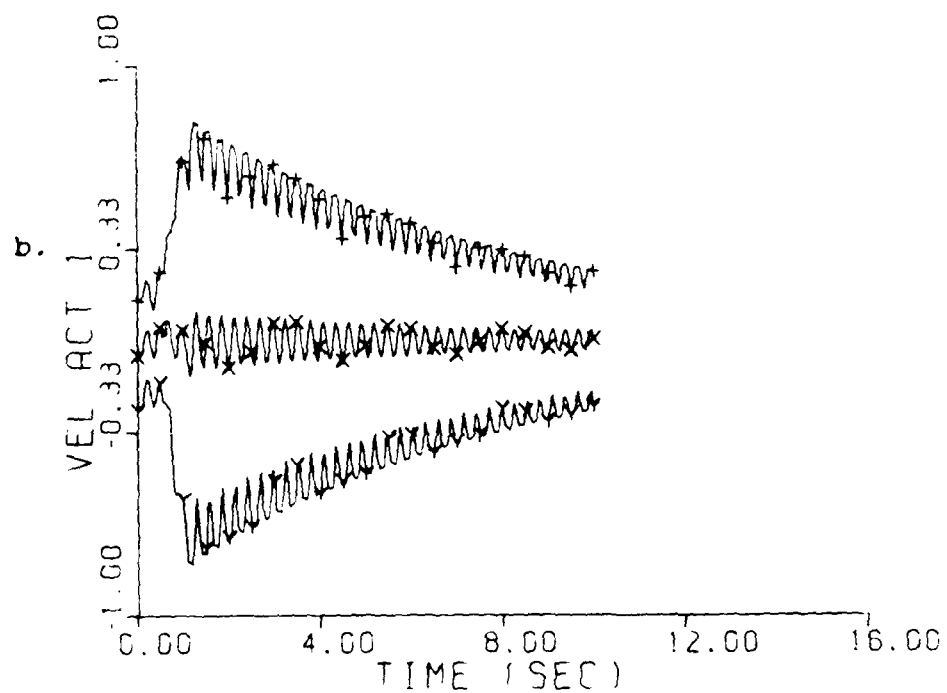
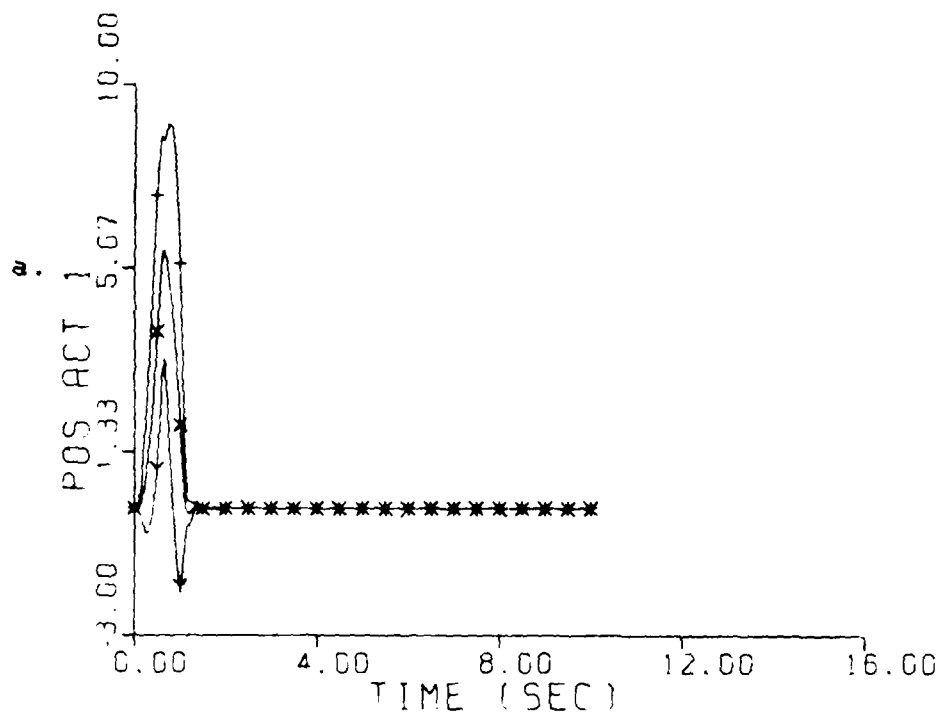


Fig. 5-26. Benchmark Controller; Full Simulation Time;
Parameter point (7,6); Mean $\pm 1\sigma$
a. Position State at node #1 (in)
b. Velocity State at node #1 (in/sec)

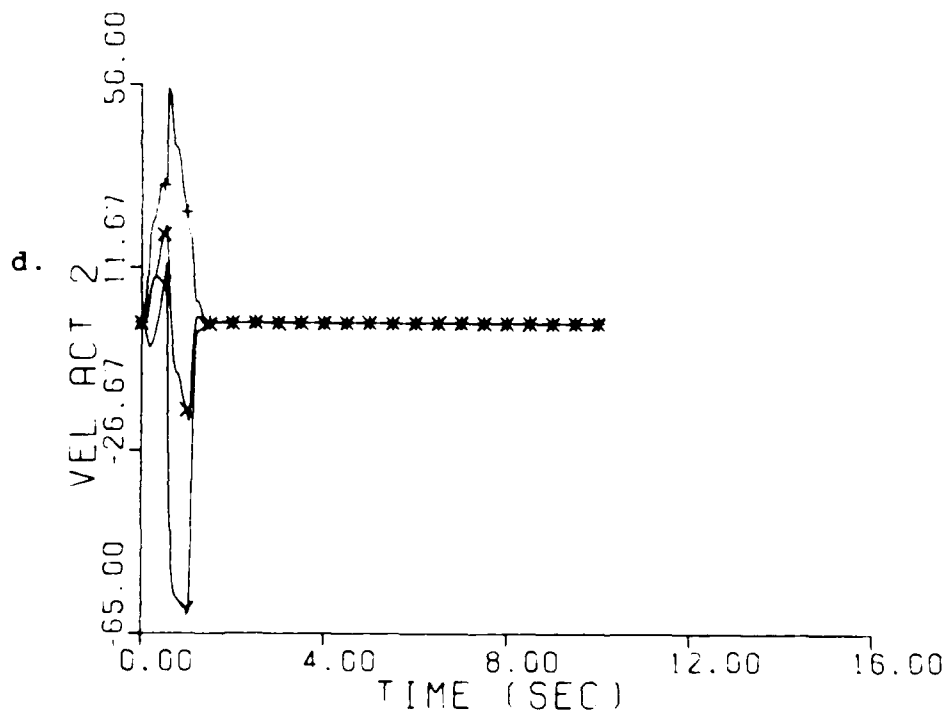
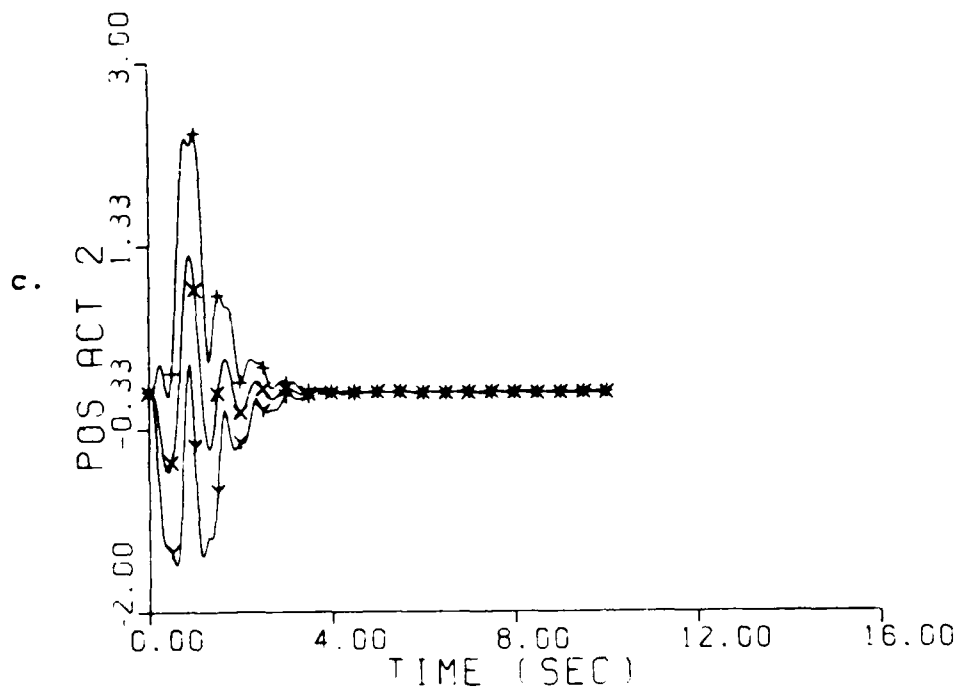


Fig. 5-26. Benchmark Controller; Full Simulation Time;
 Parameter point (7,6); Mean $\pm 1\sigma$
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

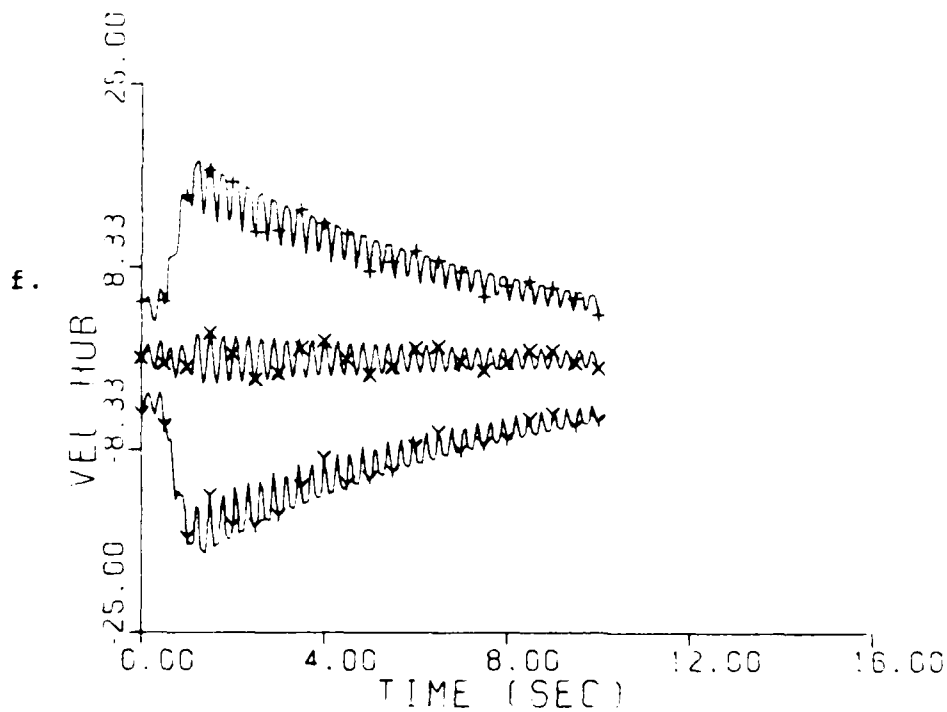
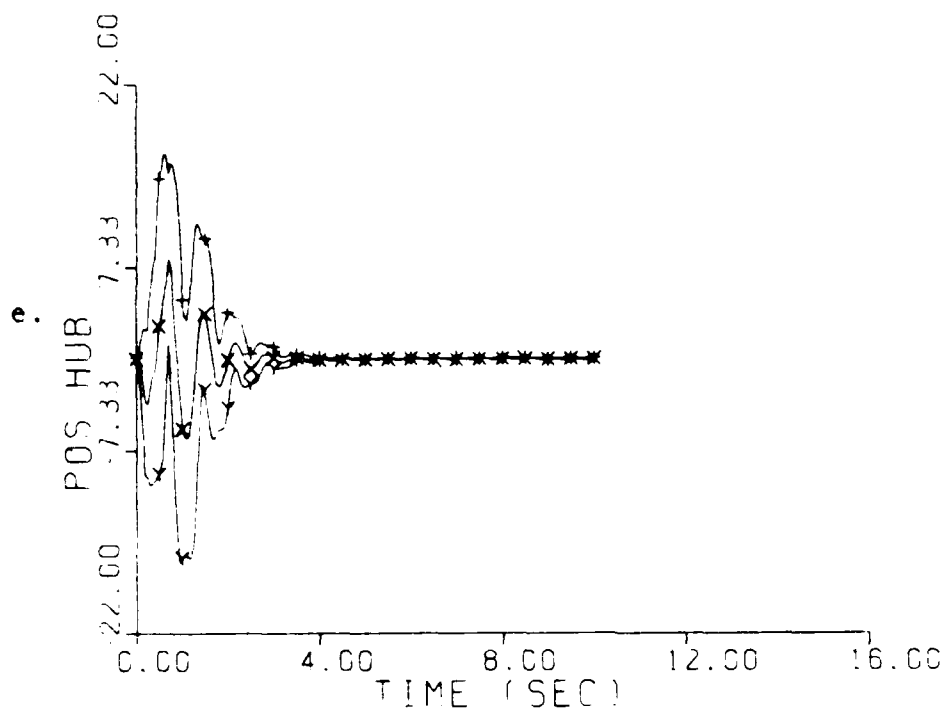


Fig. 5-26. Benchmark Controller; Full Simulation Time;
 Parameter point (7,6); Mean $\pm 1\sigma$
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

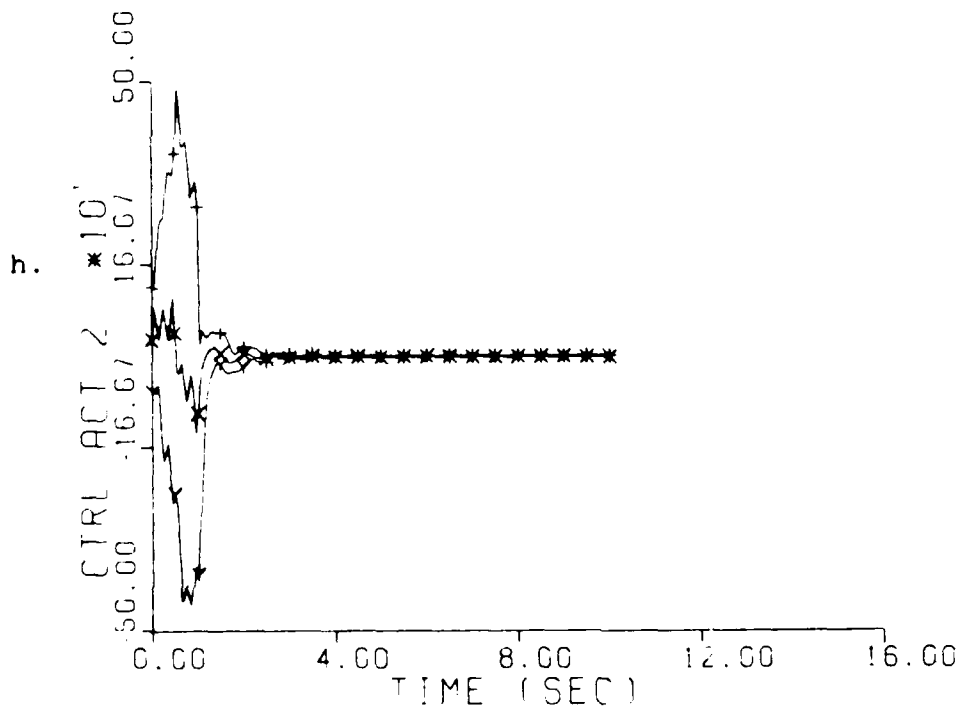
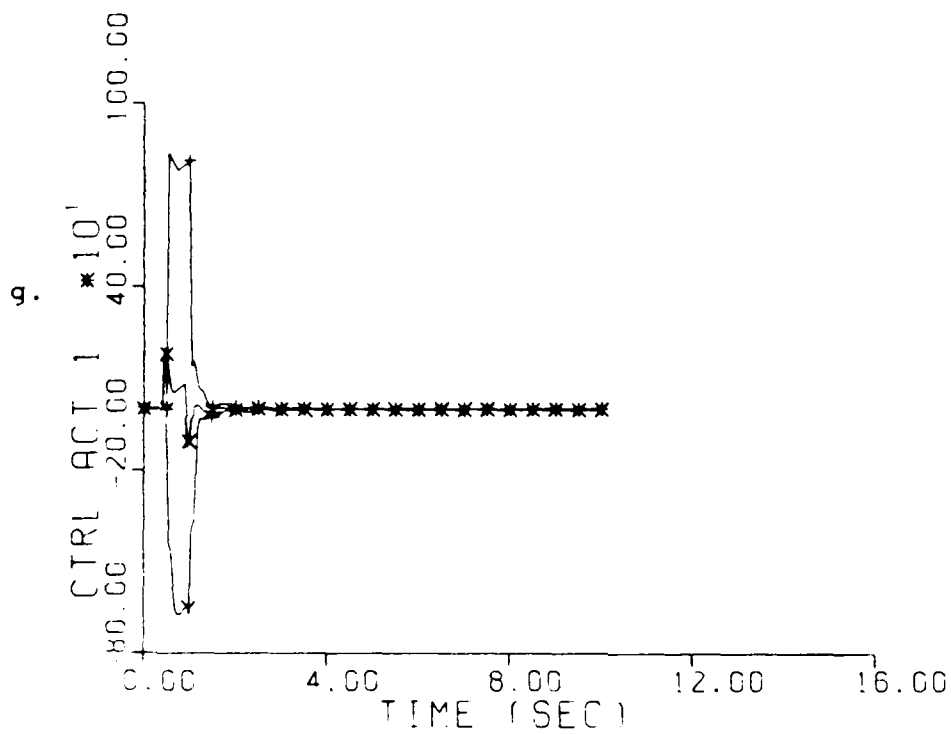


Fig. 5-26. Benchmark Controller; Full Simulation Time;
 Parameter point (7,6); Mean $\pm 1\sigma$
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

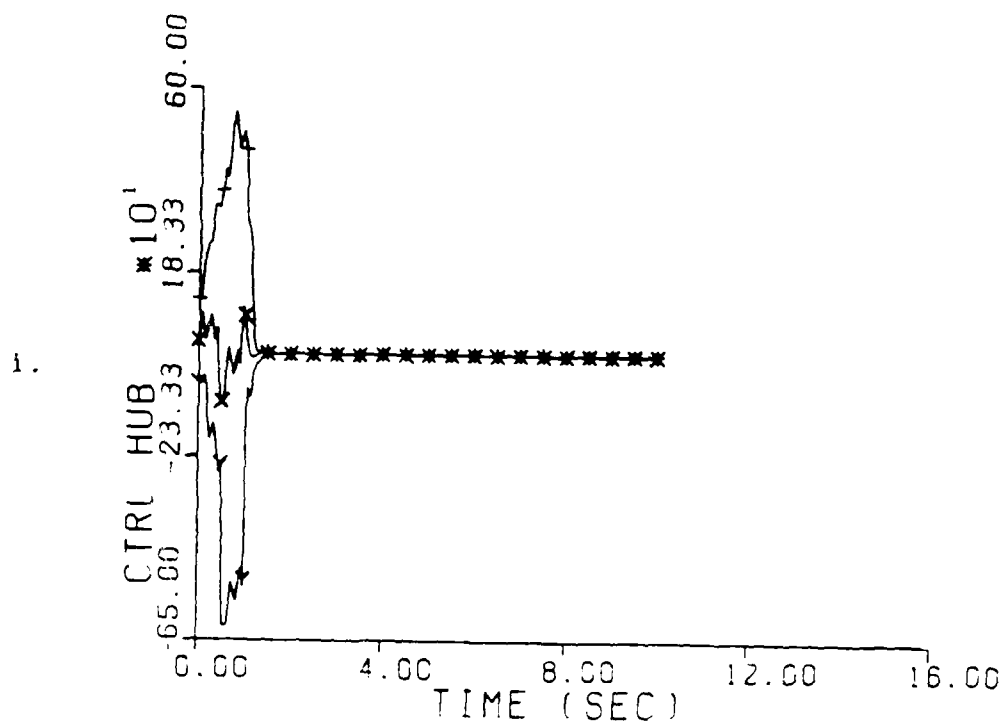


Fig. 5-26. Benchmark Controller; Full Simulation Time;
 Parameter point (7,6); Mean $\pm 1\sigma$
 i. Control Amplitude at node #3 (in)

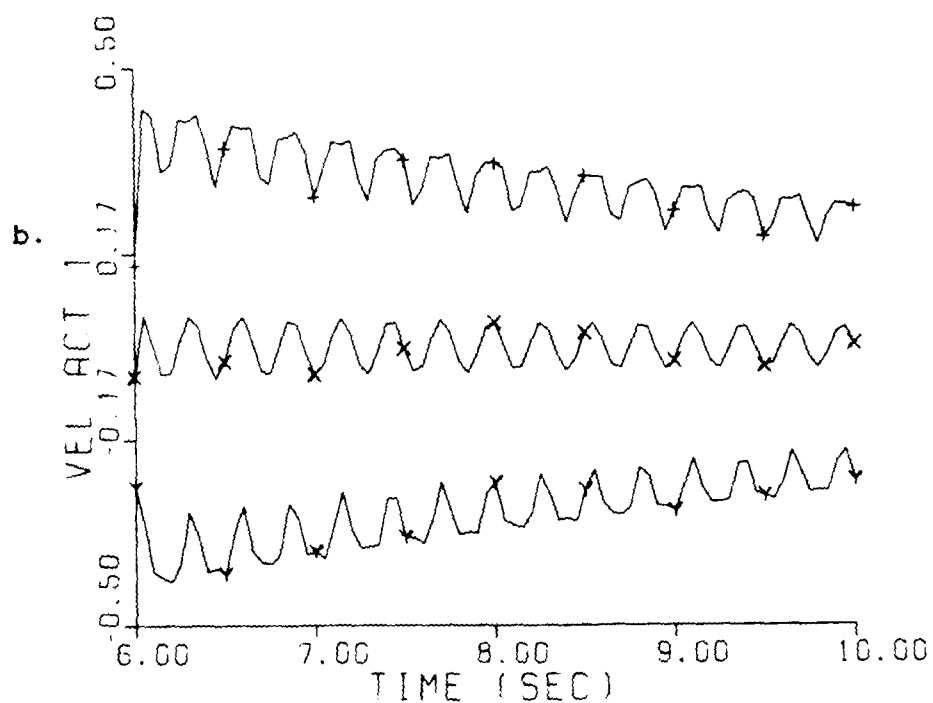
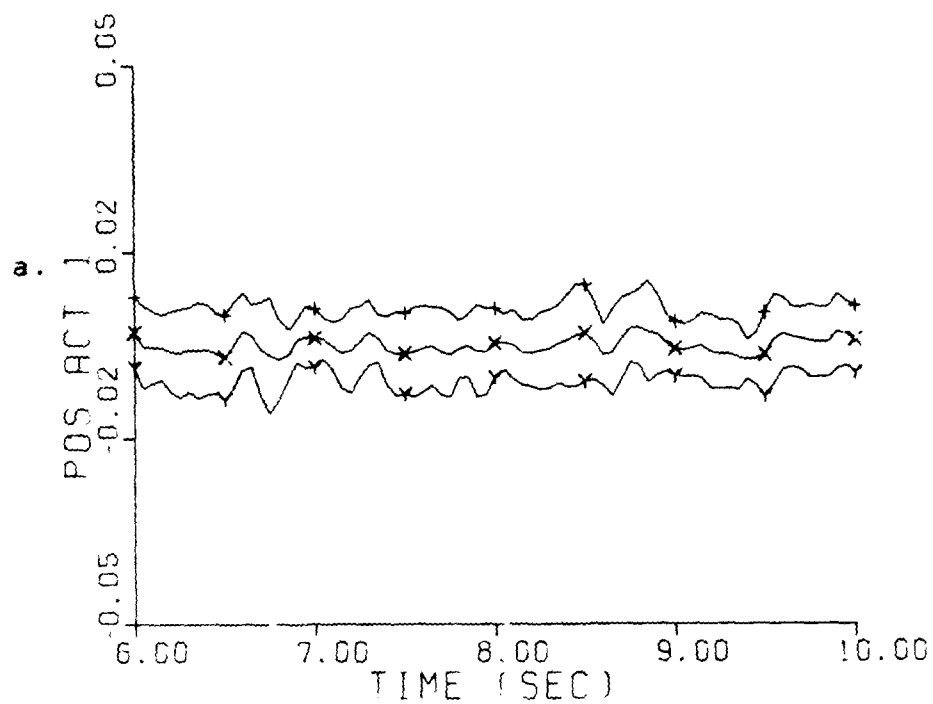


Fig. 5-27. Benchmark Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1 \sigma$
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

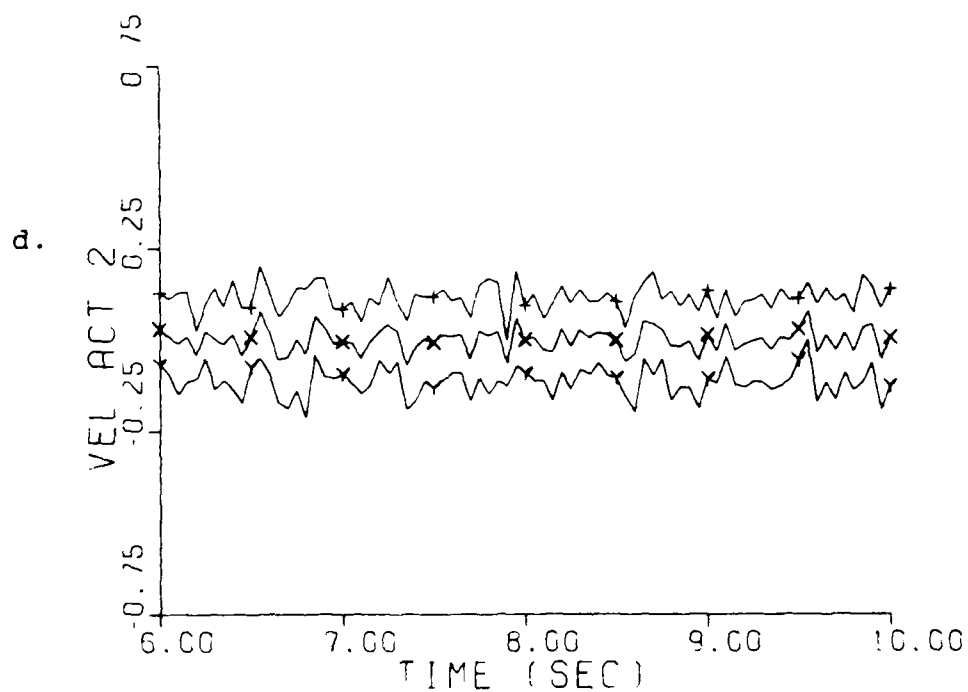
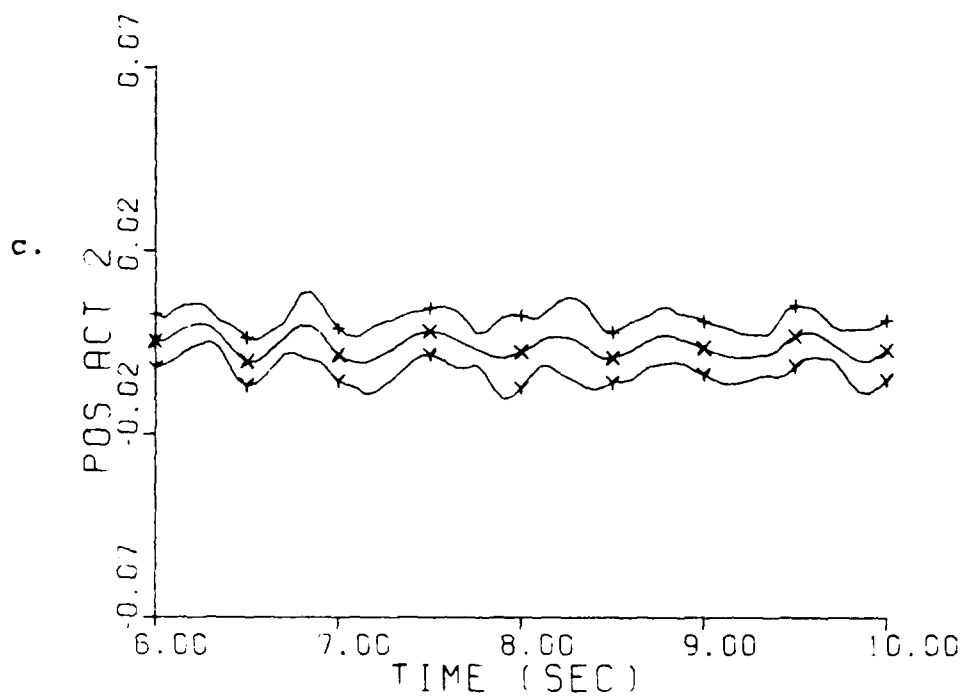


Fig. 5-27. Benchmark Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1\sigma$
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

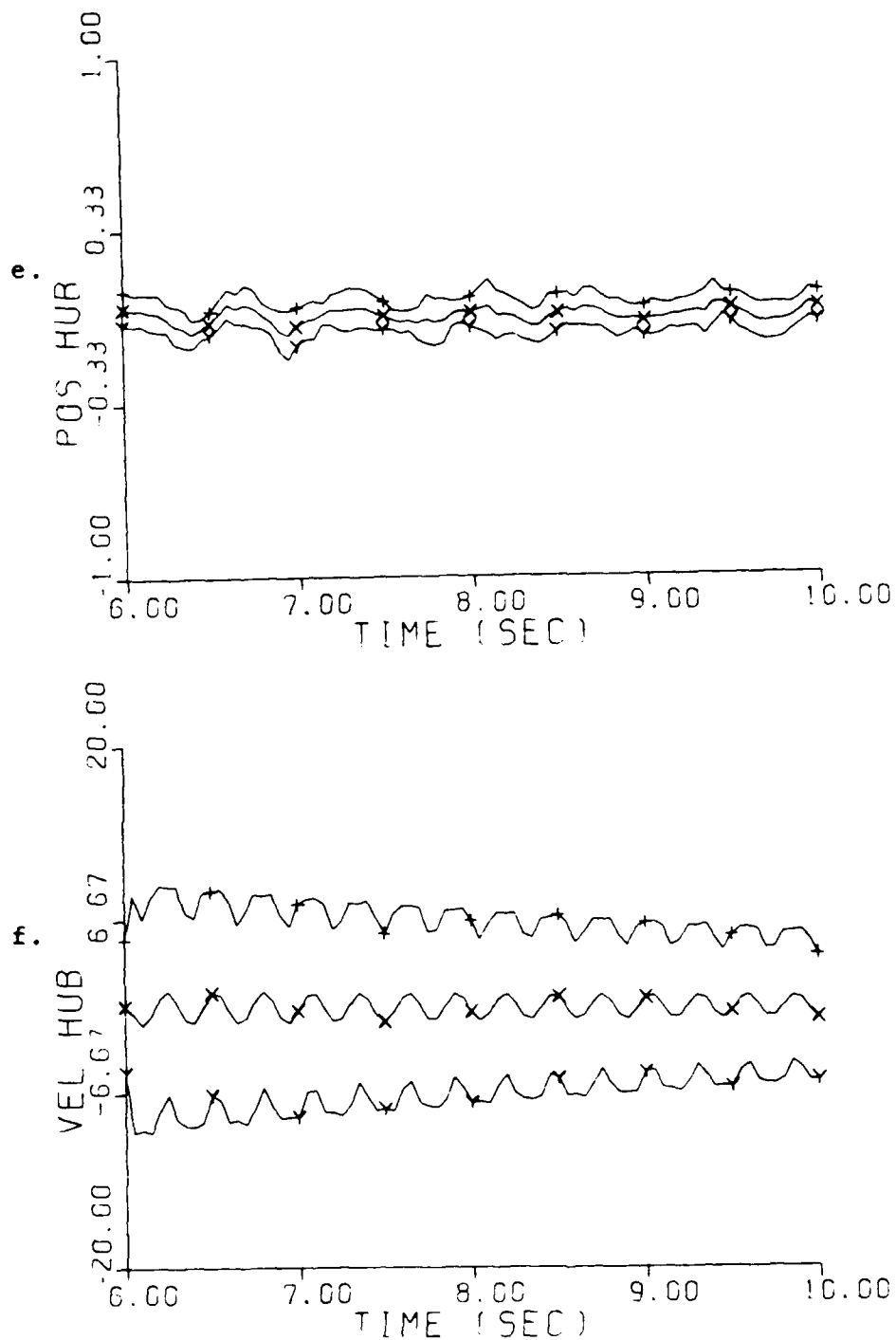


Fig. 5-27. Benchmark Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1\sigma$
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

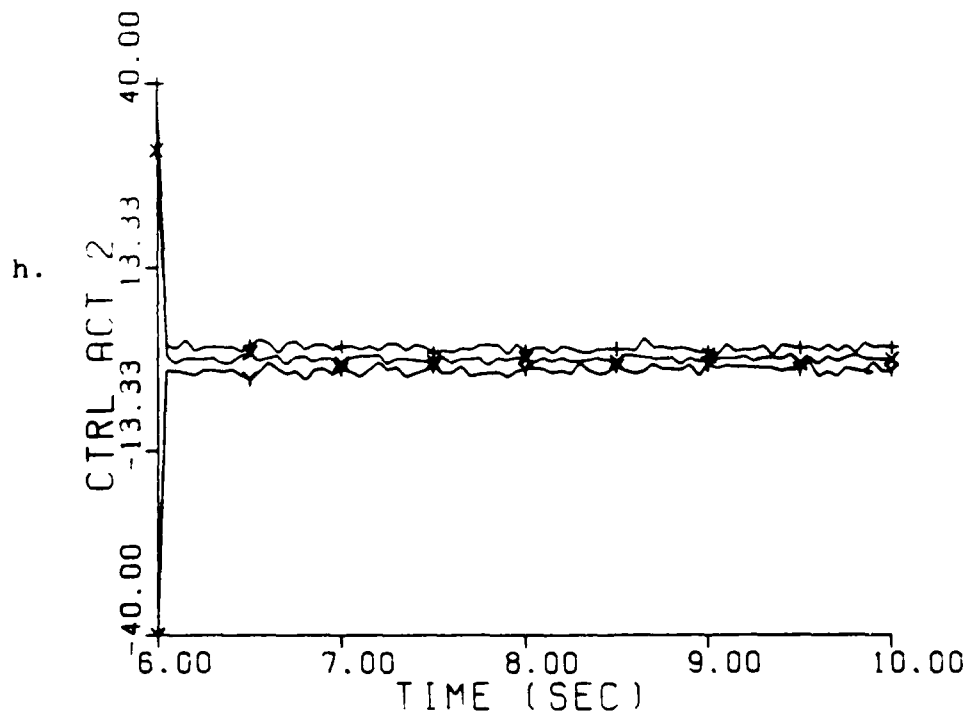
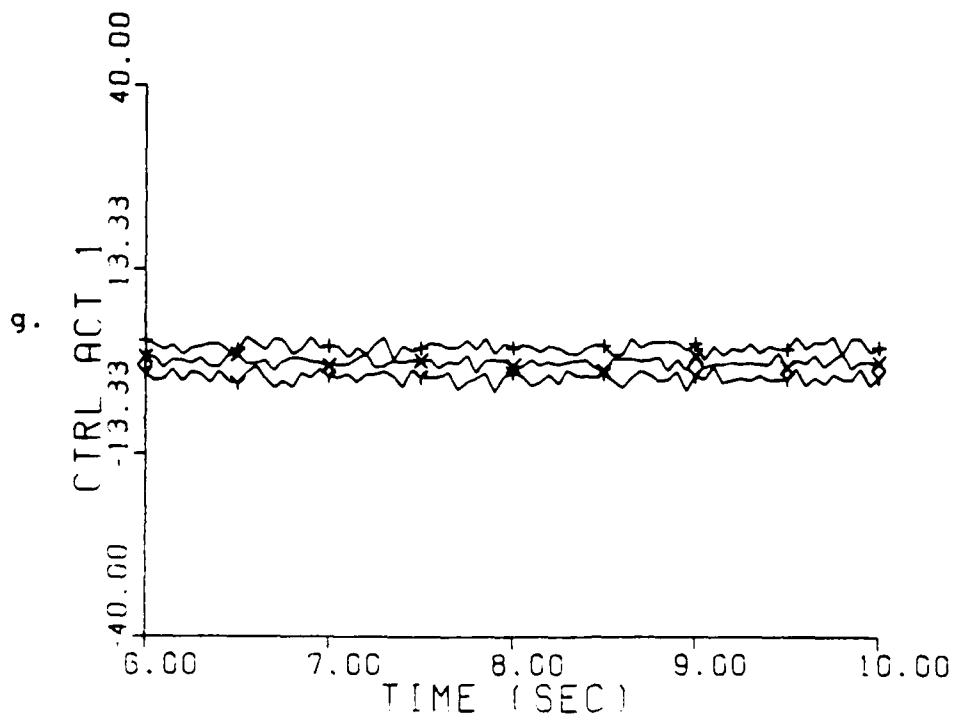


Fig. 5-27. Benchmark Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1\sigma$
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

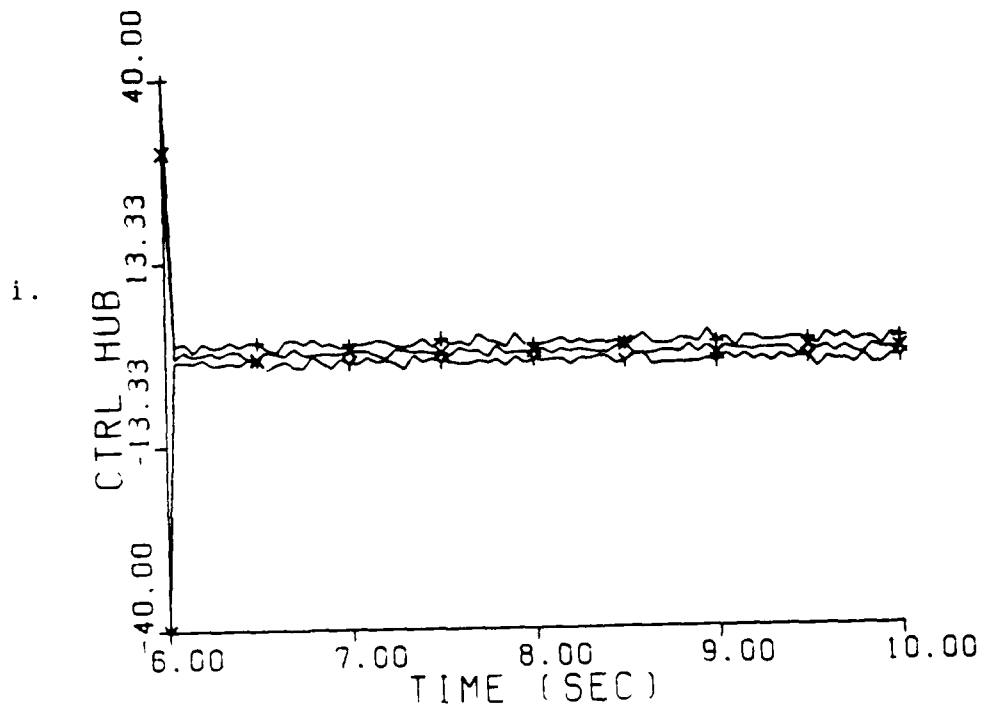


Fig. 5-27. Benchmark Controller; Post-Transient Performance
 Parameter point (7,6); Mean $\pm 1\sigma$
 i. Control Amplitude at node #3 (in)

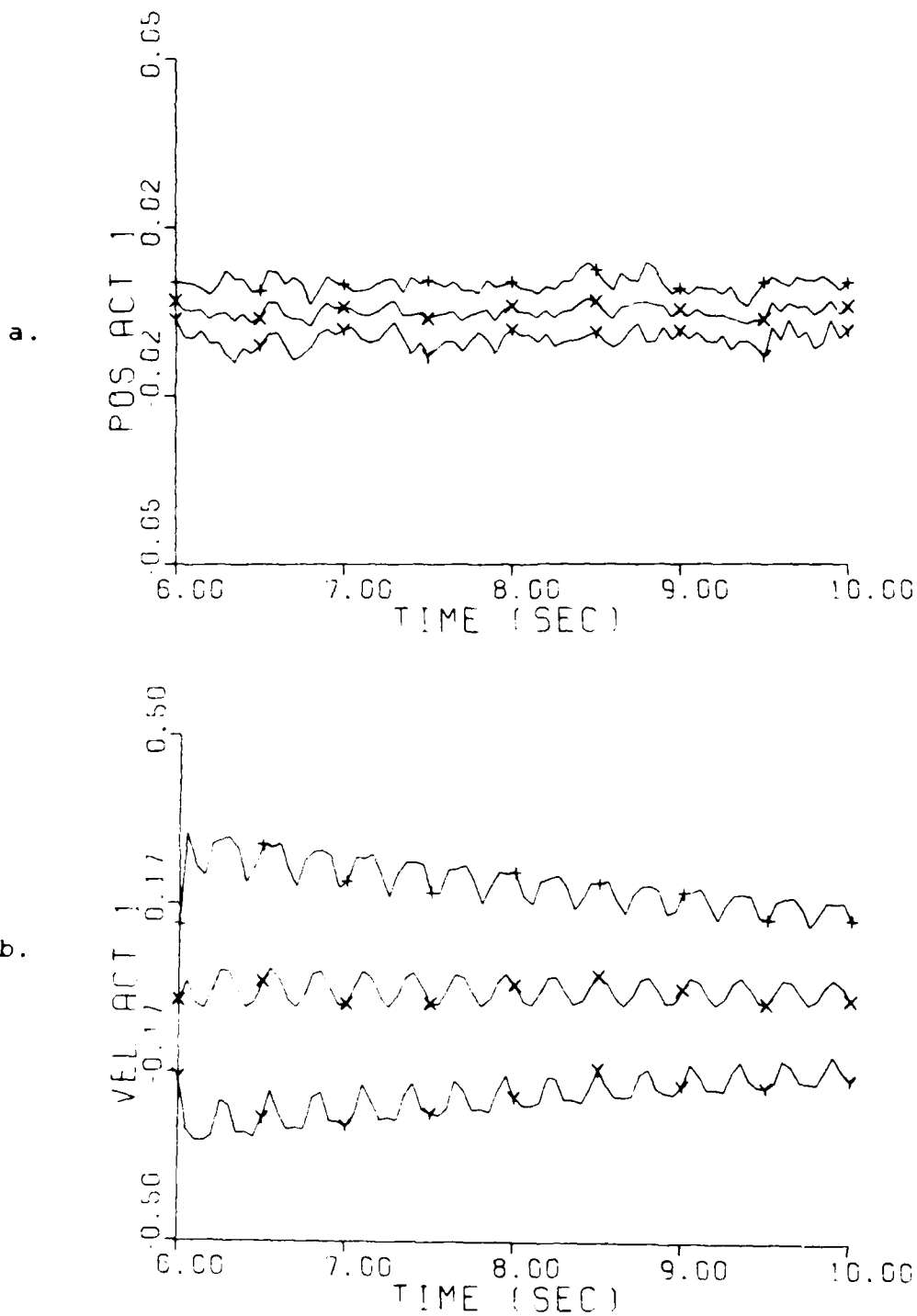


Fig. 5-28. Moving-Bank Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1 \sigma$
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

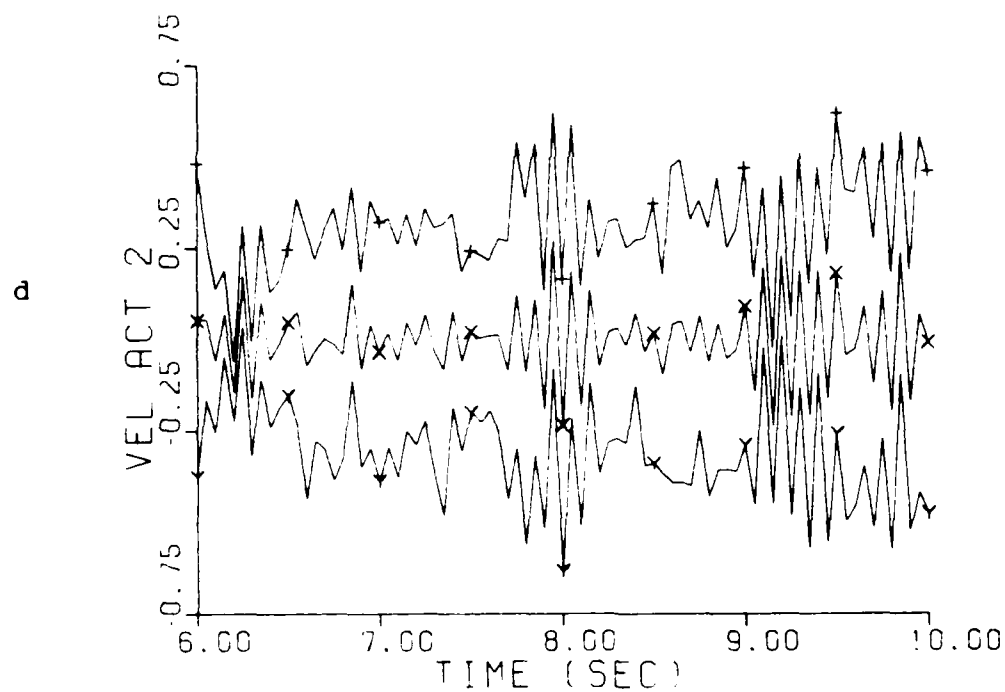
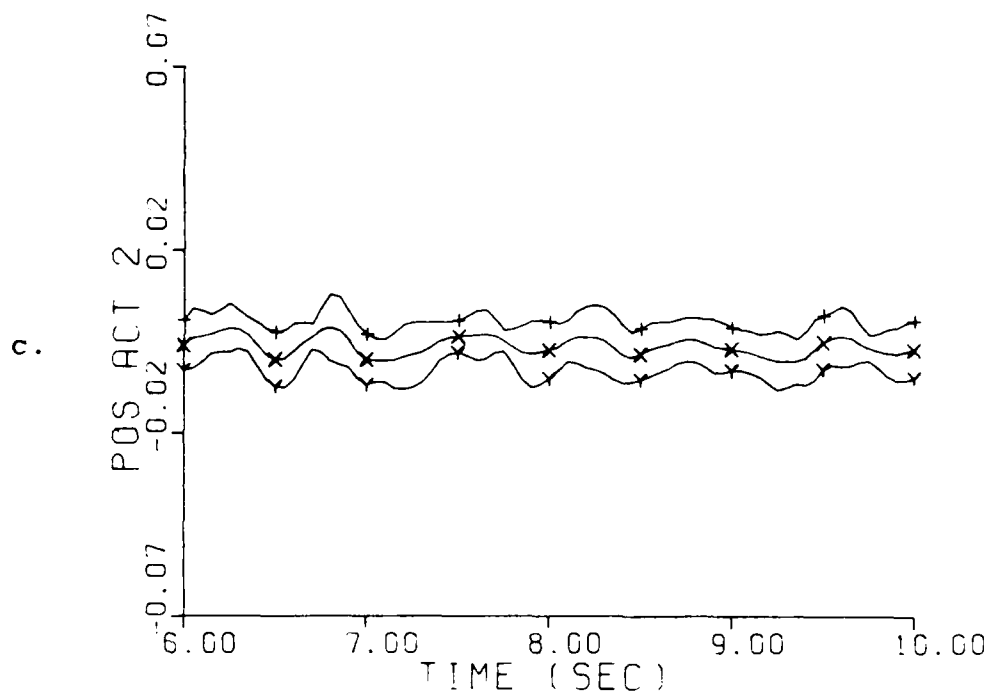


Fig. 5-28. Moving-Bank Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1 \sigma$
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

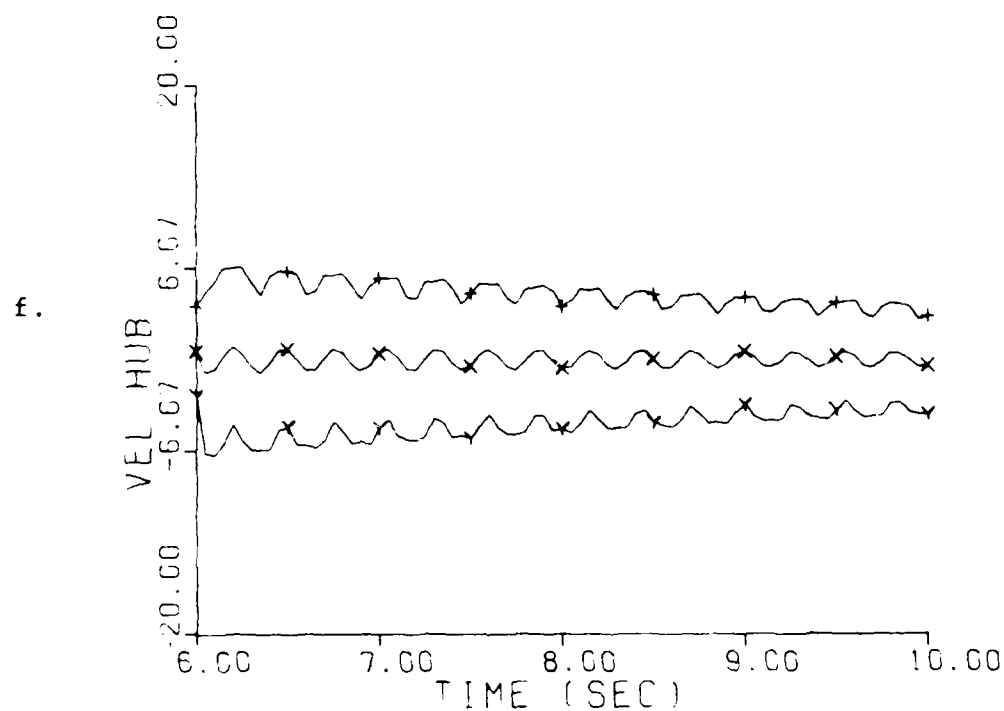
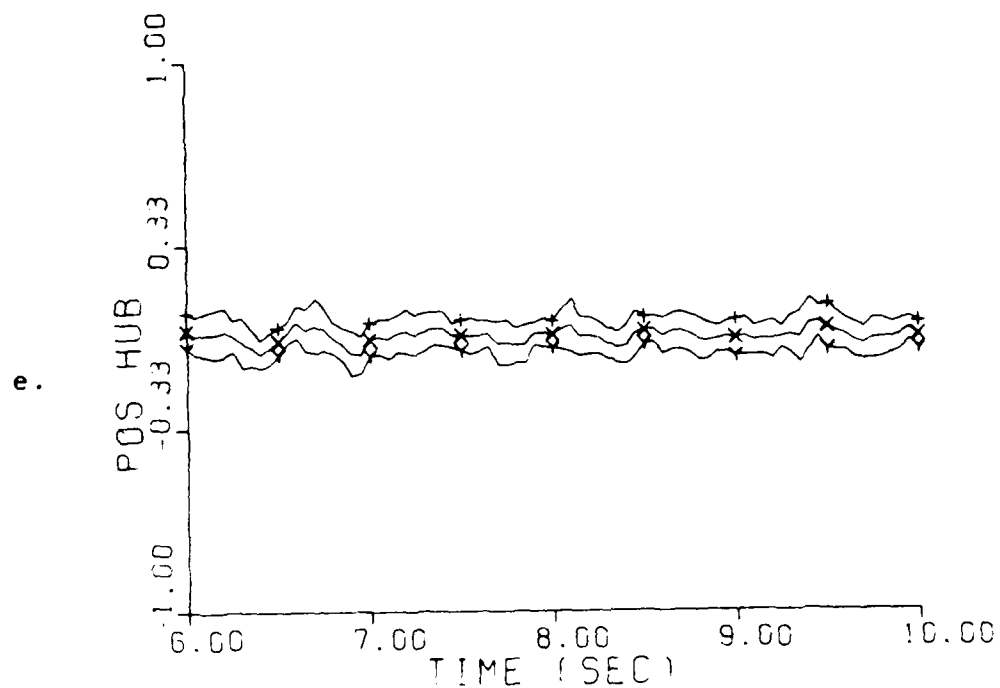


Fig. 5-28. Moving-Bank Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1 \sigma$
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

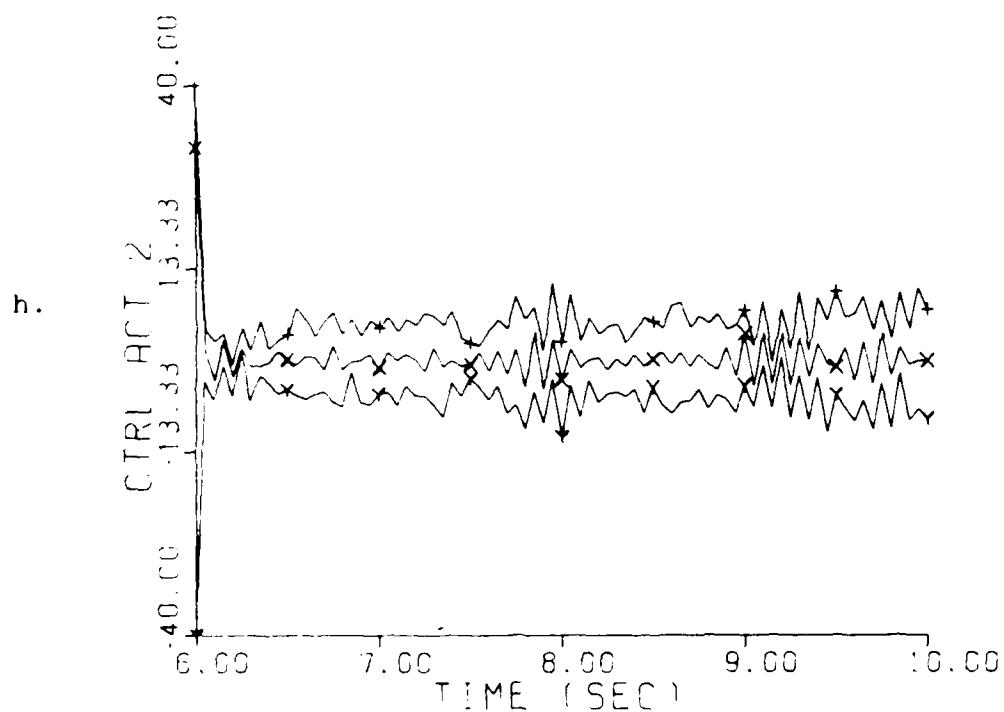
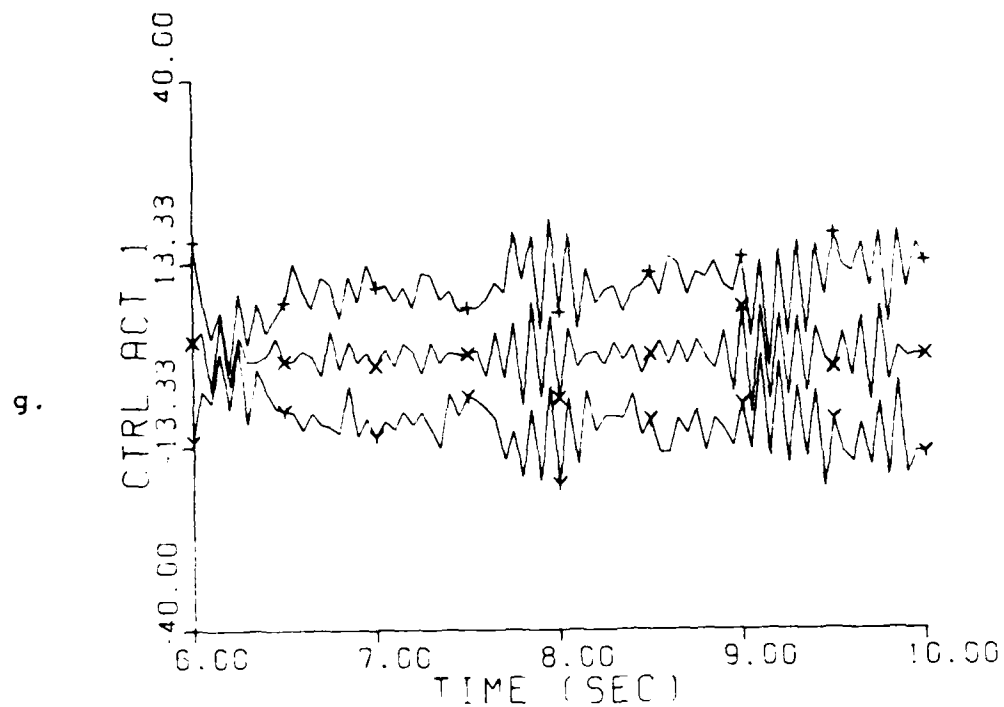


Fig. 5-28. Moving-Bank Controller; Post-Transient Performance;
 Parameter point (7,6); Mean $\pm 1\sigma$
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

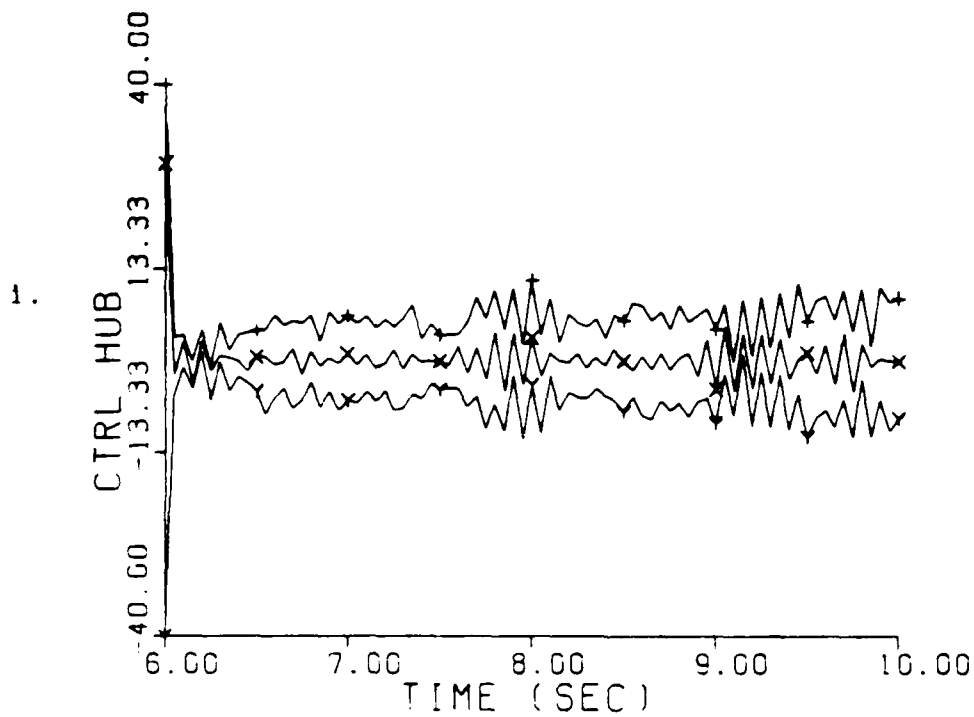


Fig. 5-28. Moving-Bank Controller; Post-Transient
Performance
Parameter point (7,6); Mean $\pm 1\sigma$
1. Control Amplitude at node #3 (in)

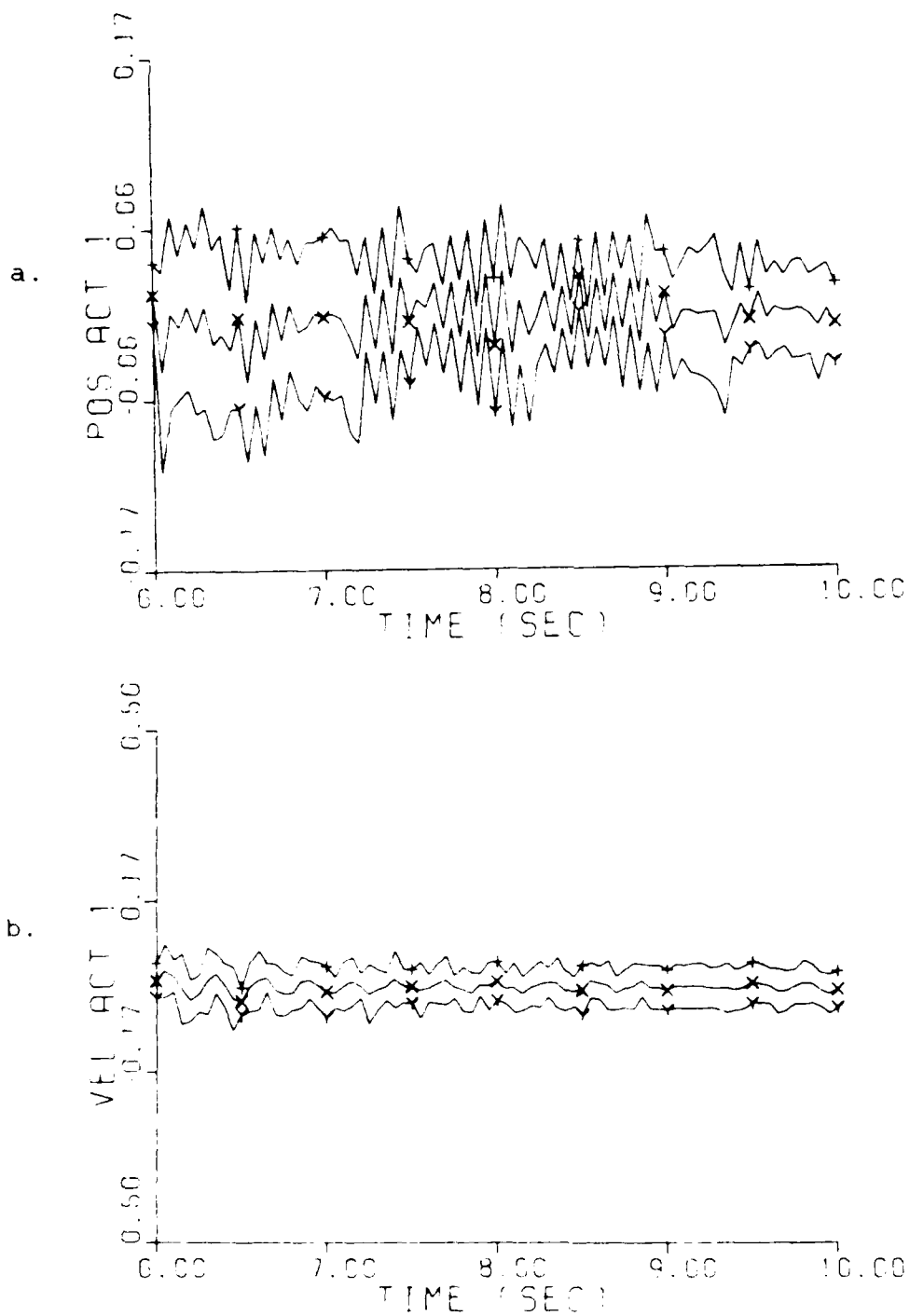


Fig. 5-29. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (7,6); Mean $\pm 1\sigma$
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

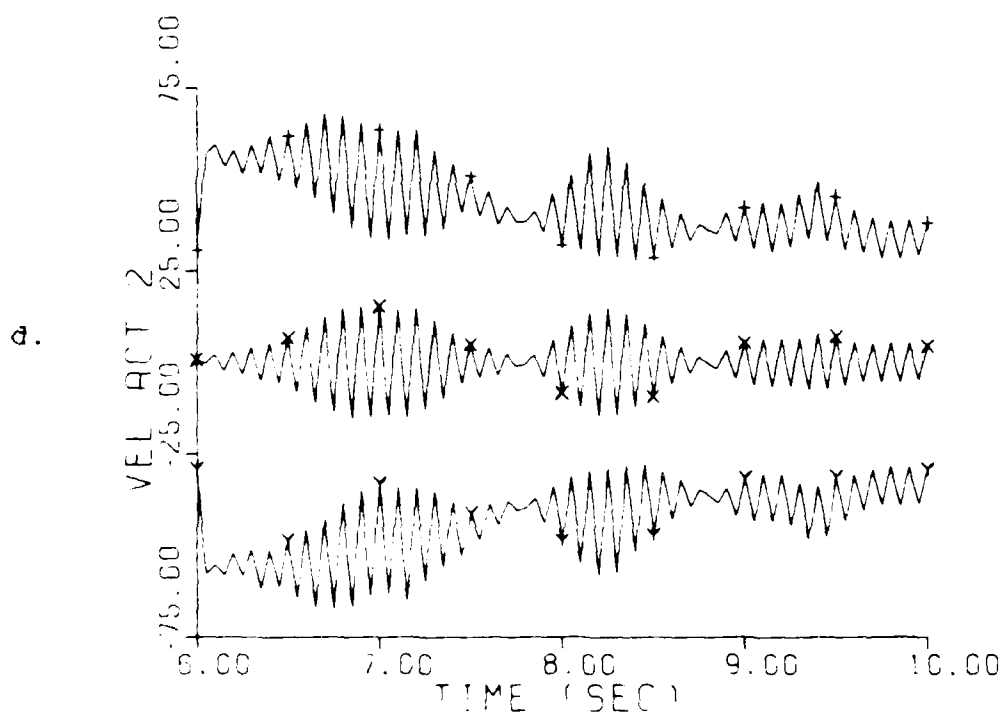
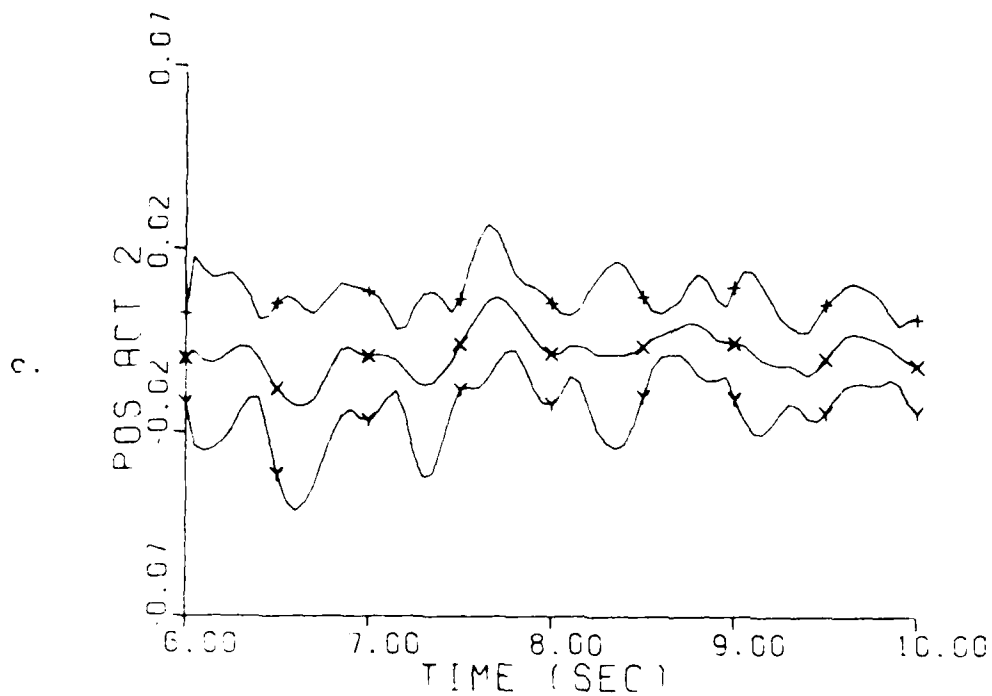


Fig. 5-29. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (7,6); Mean +1 σ
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

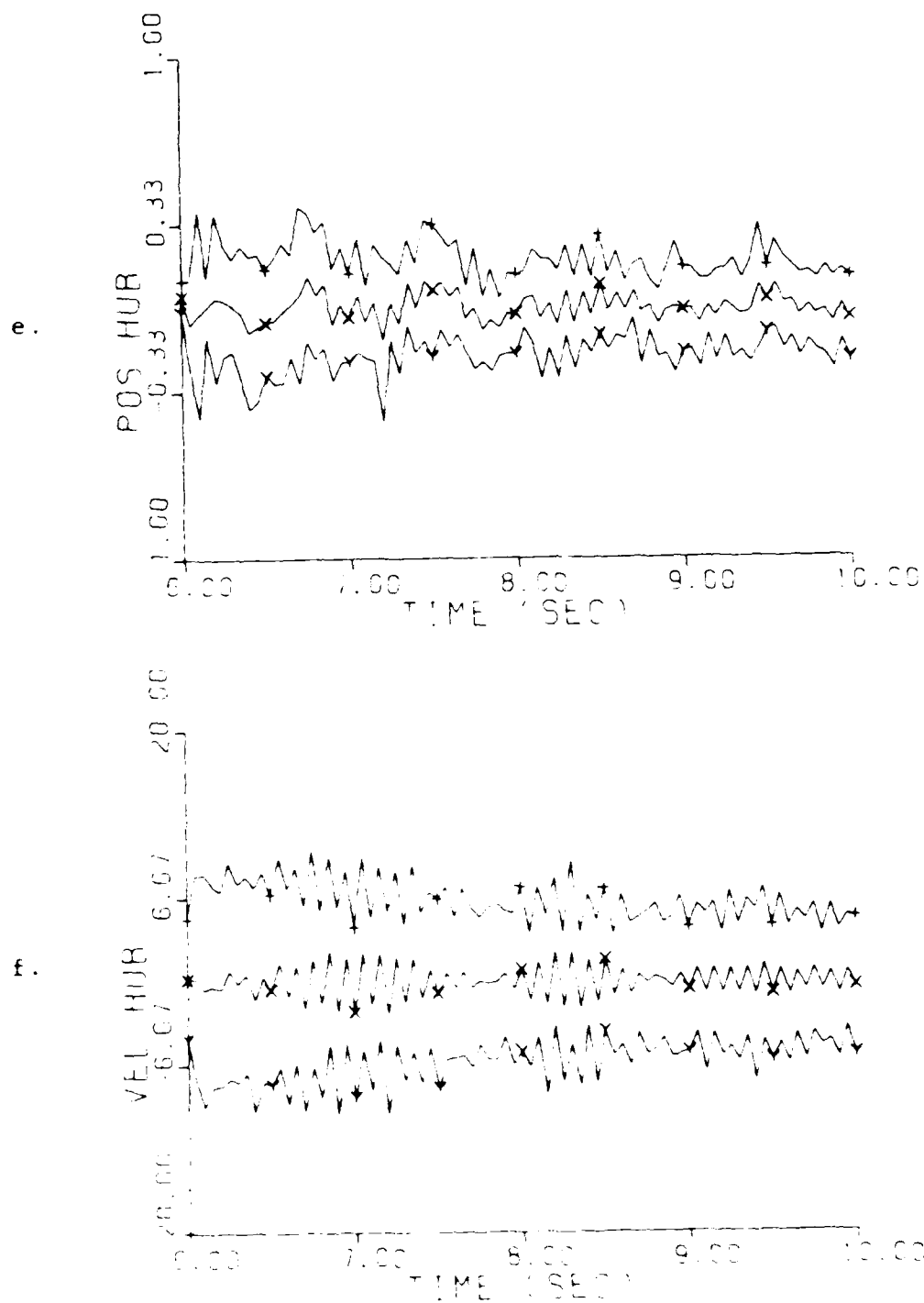


Fig. 5-29. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (7,6); Mean $\pm 1\sigma$
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

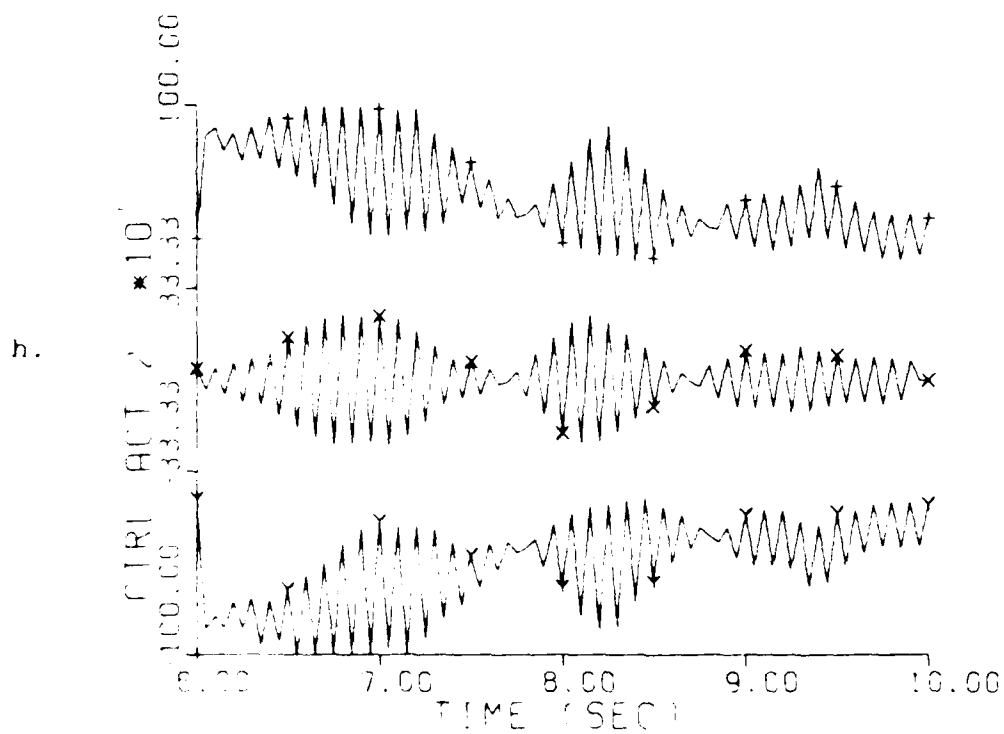
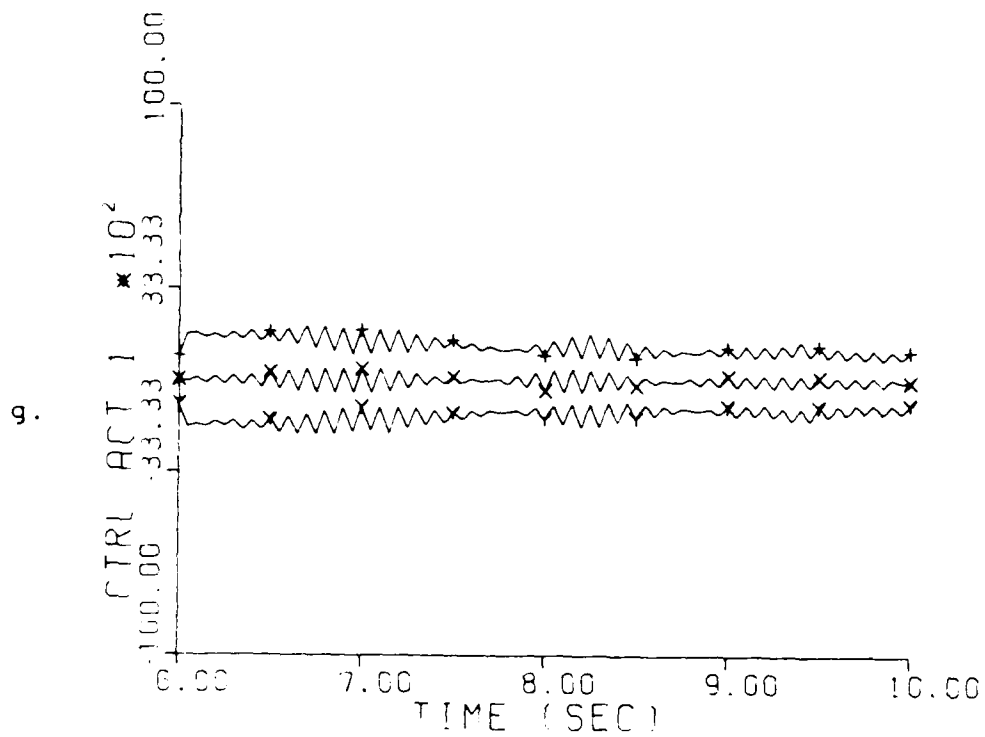


Fig. 5-29. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (7,6); Mean $+10$
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

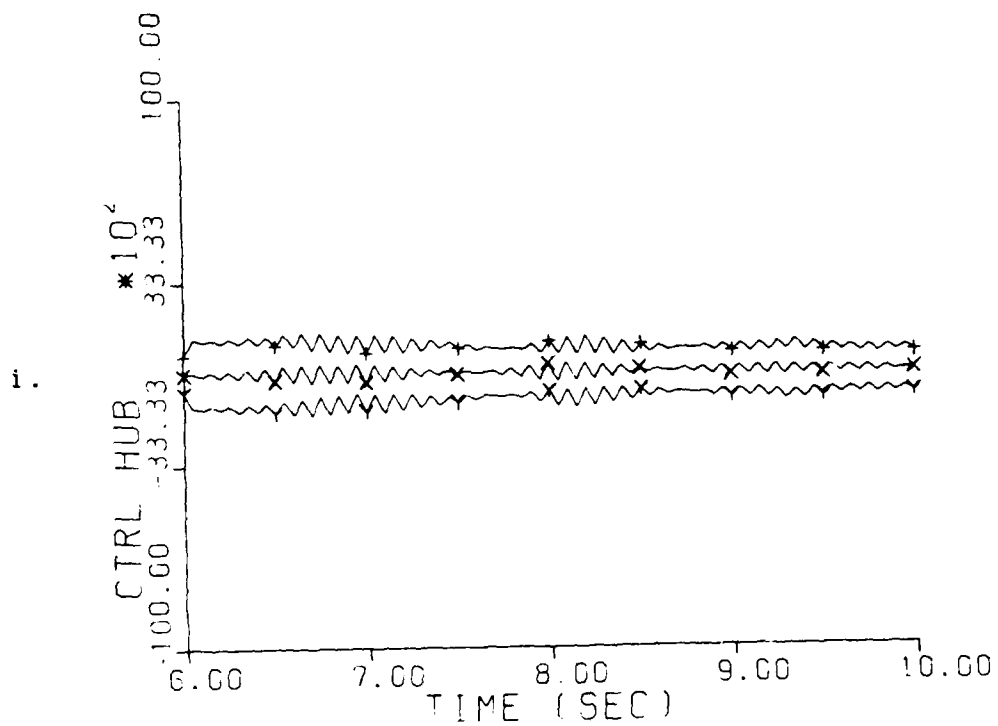


Fig. 5-29. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (7,6); Mean ± 10
 i. Control Amplitude at node #3 (in)

started at $t = 6.0$ sec in order to remove the initial large-amplitude transient effects. Also recall that the center of the moving bank is always started at the parameter point (5,5), which corresponds to the center of the parameter space. Figure 5-27 contains the plots for the benchmark controller. Figure 5-28 contains the plots for the moving-bank controller, and Figure 5-29 contains the plots for the fixed-bank controller. The most important states to compare are the position states because these states give an indication of movement of the truss and it is this movement that the controller is to quell to zero. The figures show that the moving-bank and the benchmark controllers performed nearly identically for the three position states. The similarity between the benchmark controller and the moving-bank controller was expected because the moving-bank algorithm estimate of the true parameter locked onto the actual true parameter value. After about 0.5 sec, 99.9 percent of the probability weighting is being applied to the filter that corresponds to the true parameter value (i.e., center of the bank), therefore the moving-bank algorithm is performing essentially the same as the benchmark controller. There were some differences however between the moving-bank controller and the benchmark controller. The benchmark controller performed much better for the state corresponding to the velocity at

node 2, although more importantly, the benchmark controller required much less control energy than the moving-bank controller. This is probably due to the fact that the moving-bank controller does have to go through some transient effects before the estimate of the true parameter locks onto the true parameter. Note also, that the velocity states are not quite in steady state. However, this does not pose a problem because the figures clearly show that the velocity states are approaching steady state condition. For this set of simulation runs the fixed-bank controller performed (see Figure 5-29) much worse than the benchmark controller. The reason for this poor performance is two-fold. First, it was initially assumed that the fixed-bank controller would put most of its probability weight on the four filters that surround the true parameter point. Instead, it put nearly all of its probability weight (i.e. 1) on one filter and bounced this heavy weight factor nonsymmetrically throughout the nine filters in the fixed bank configuration. In addition, the X and U matrices were tuned for the (7,6) parameter point. Therefore, the locations in parameter space where the fixed-bank controller is placing its heavy weighting is not tuned for the X and U matrices.

To show this need for tuning X and U matrices for every parameter point, Figures 5-30 and 5-31 were

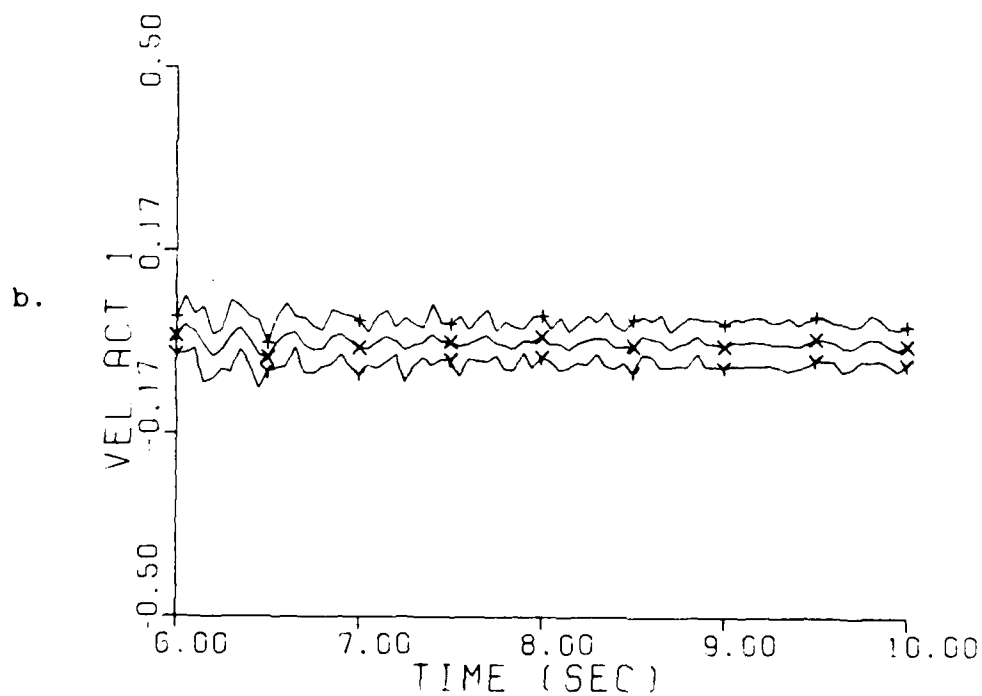
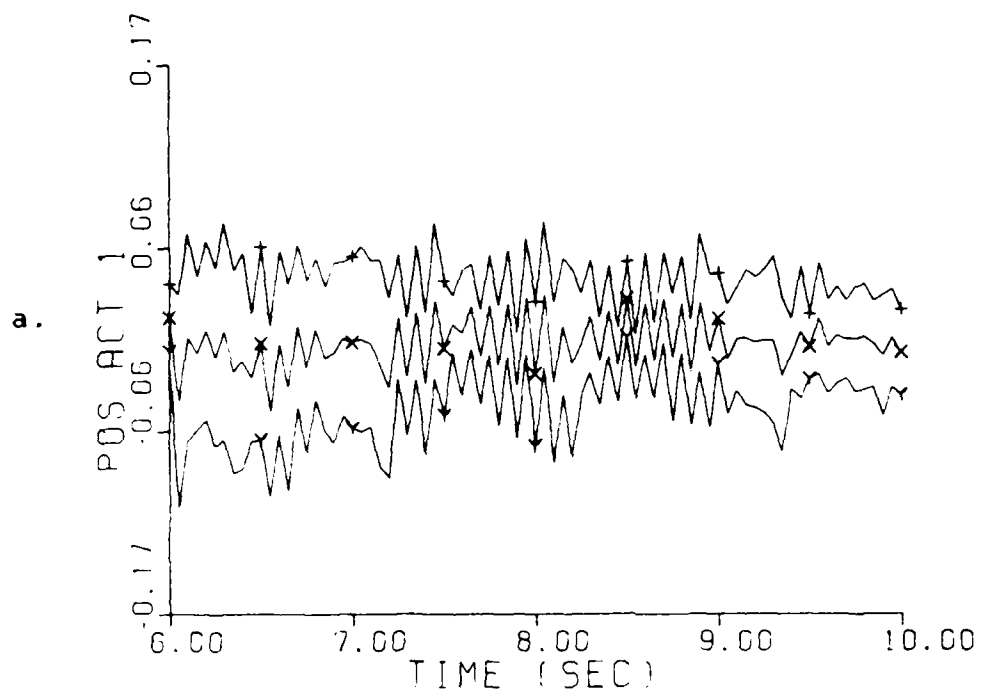


Fig. 5-30. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

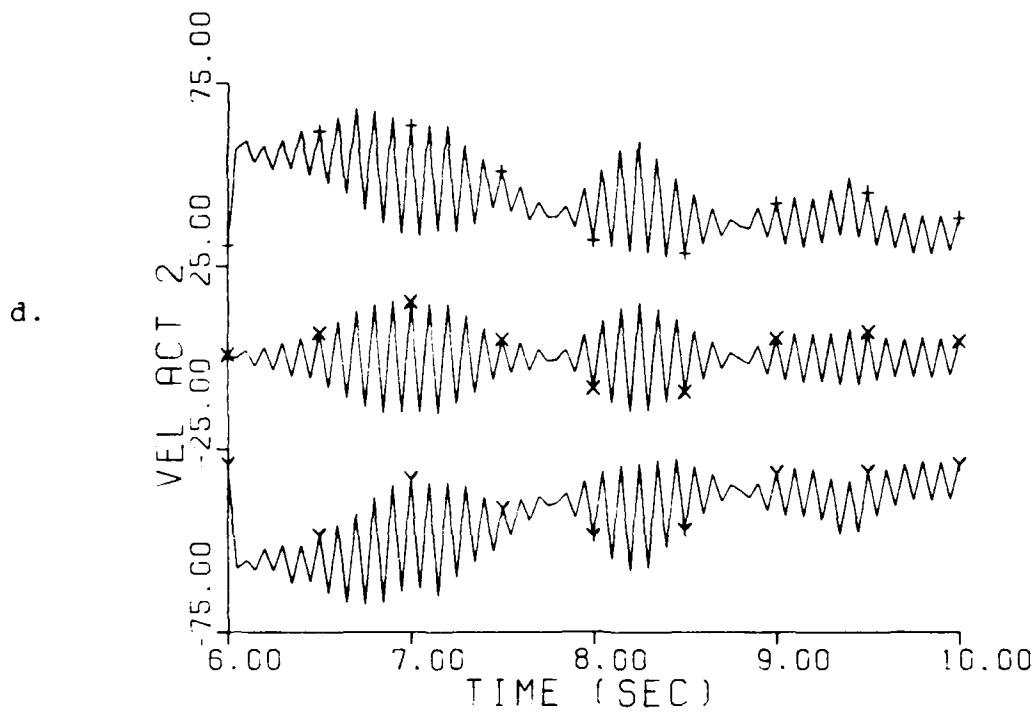
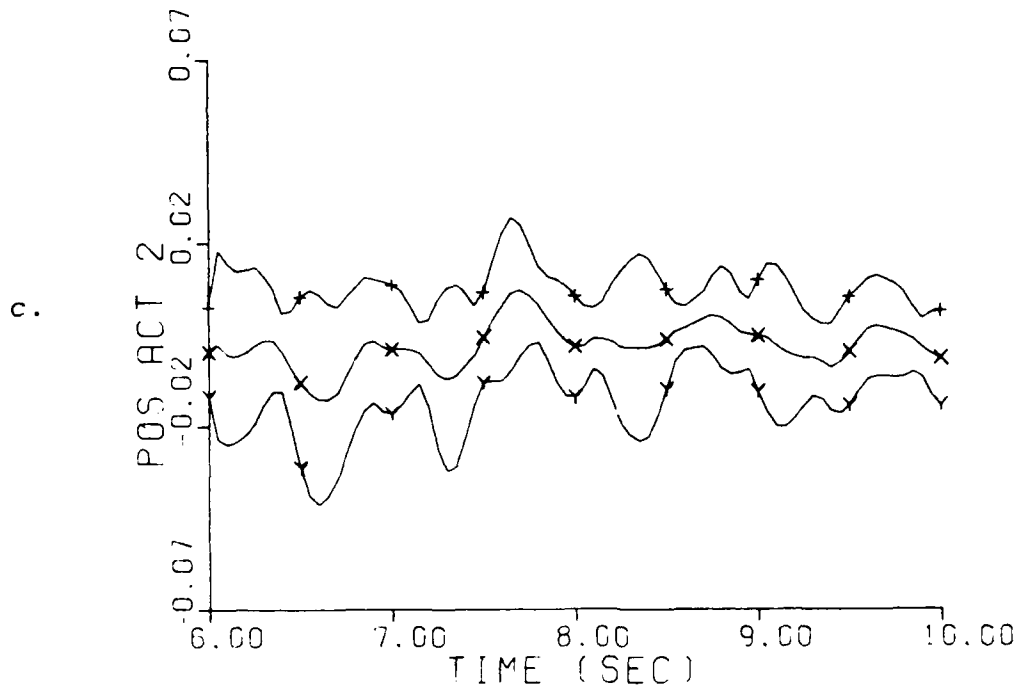


Fig. 5-30. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

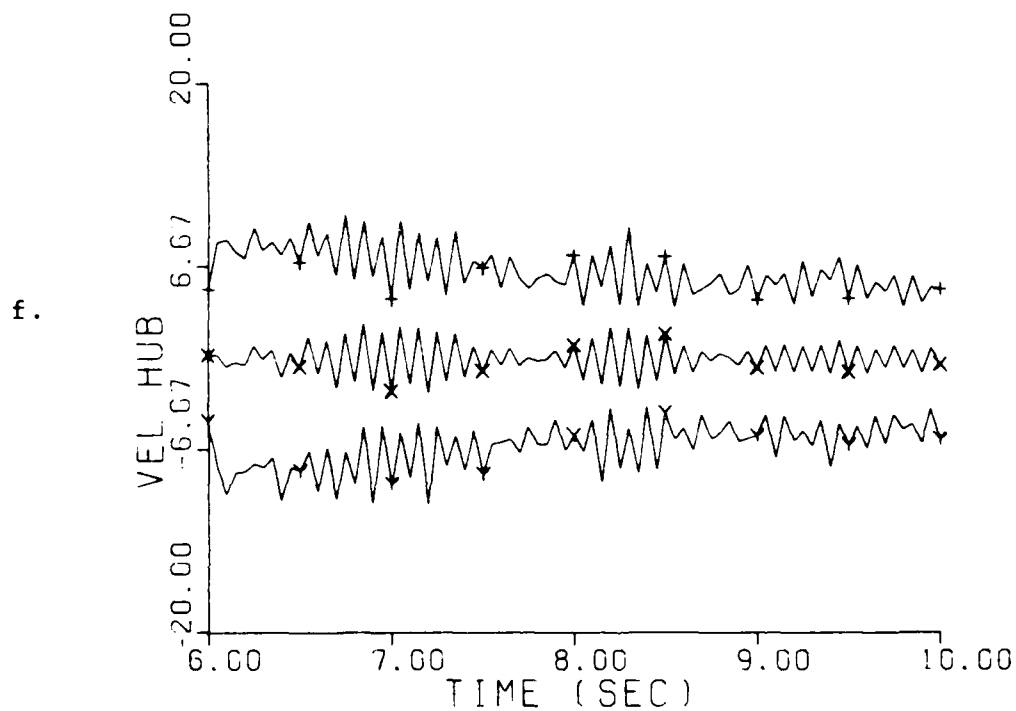
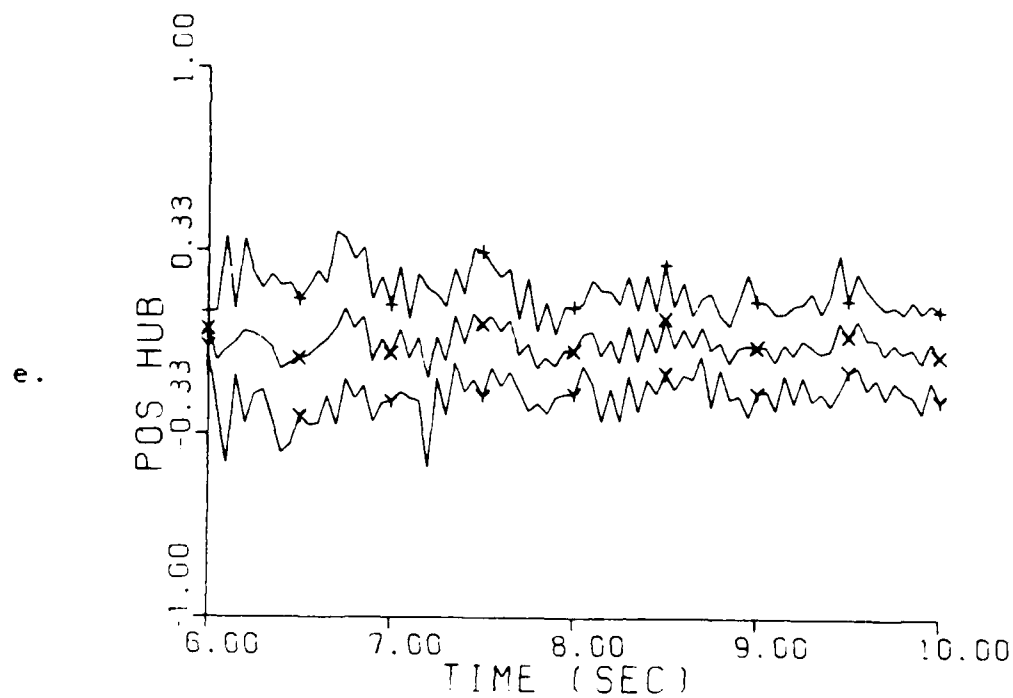


Fig. 5-30. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

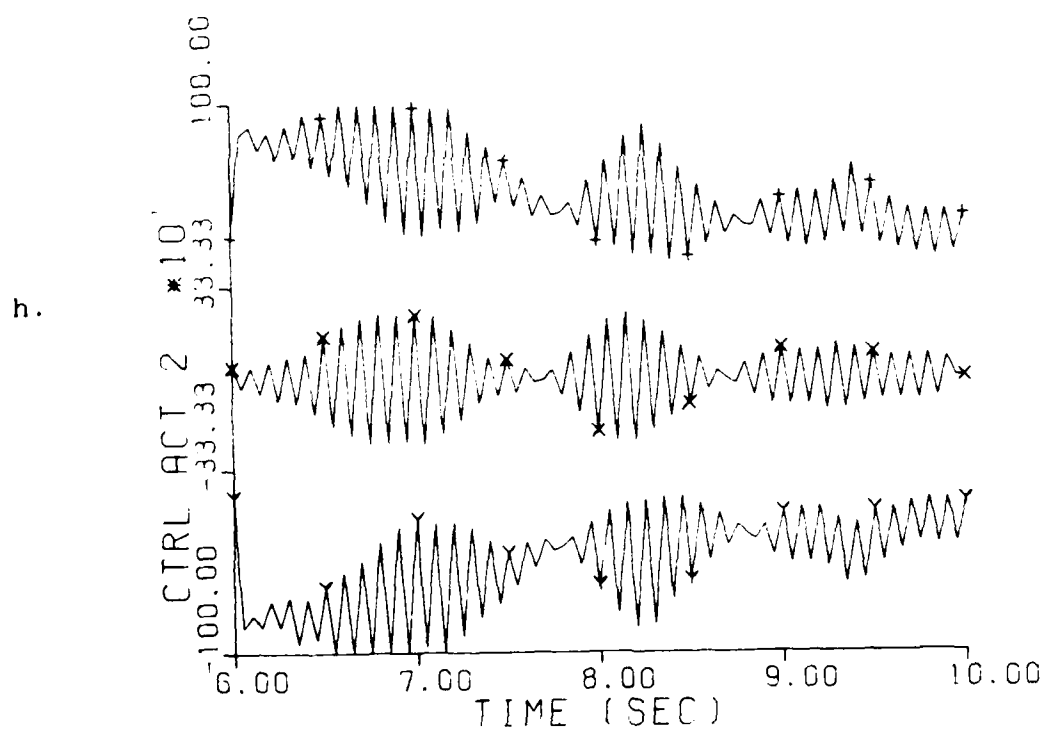
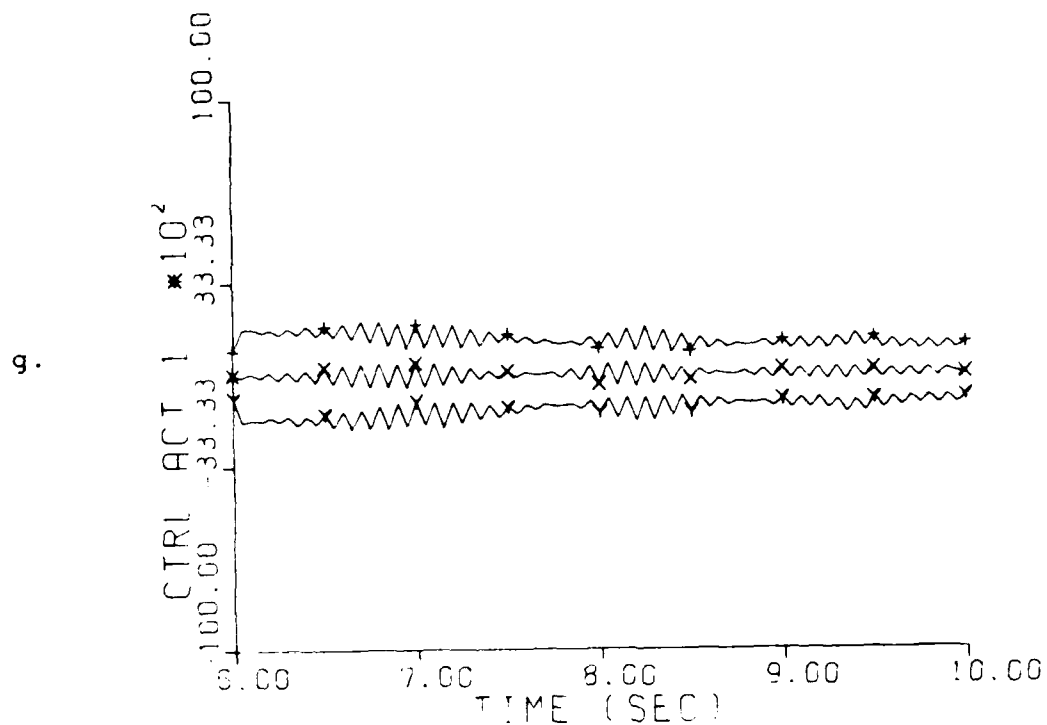


Fig. 5-30. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

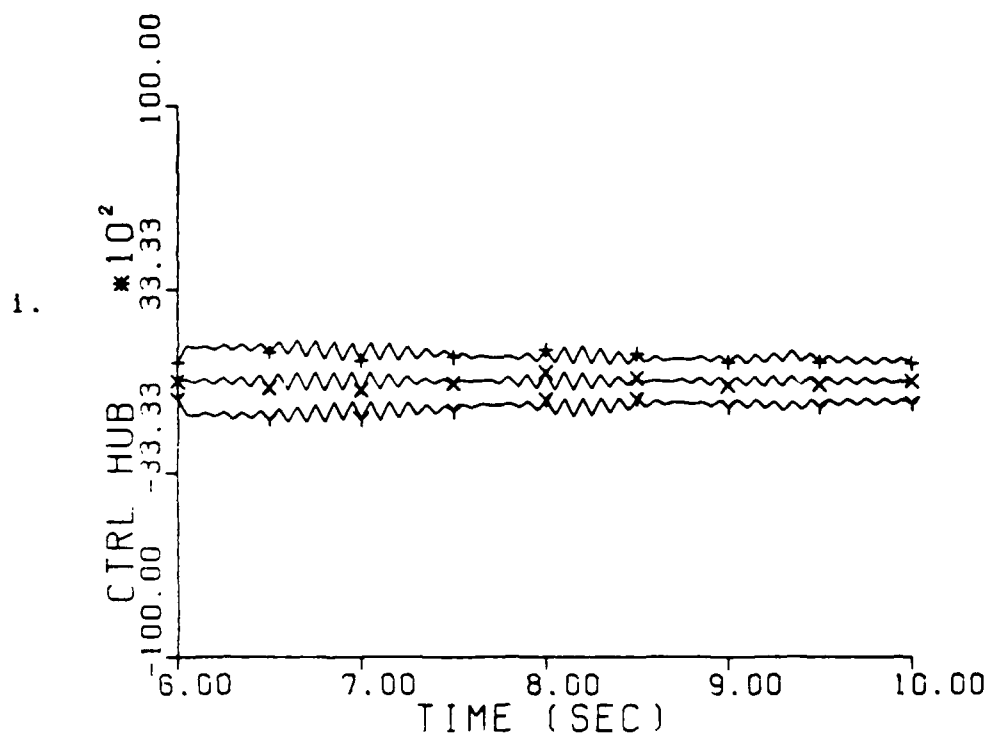


Fig. 5-30. Fixed-Bank Controller; Post-Transient
Performance
Parameter point (5,5); Mean $\pm 1 \sigma$
X and U Mistuned
1. Control Amplitude at node #3 (in)

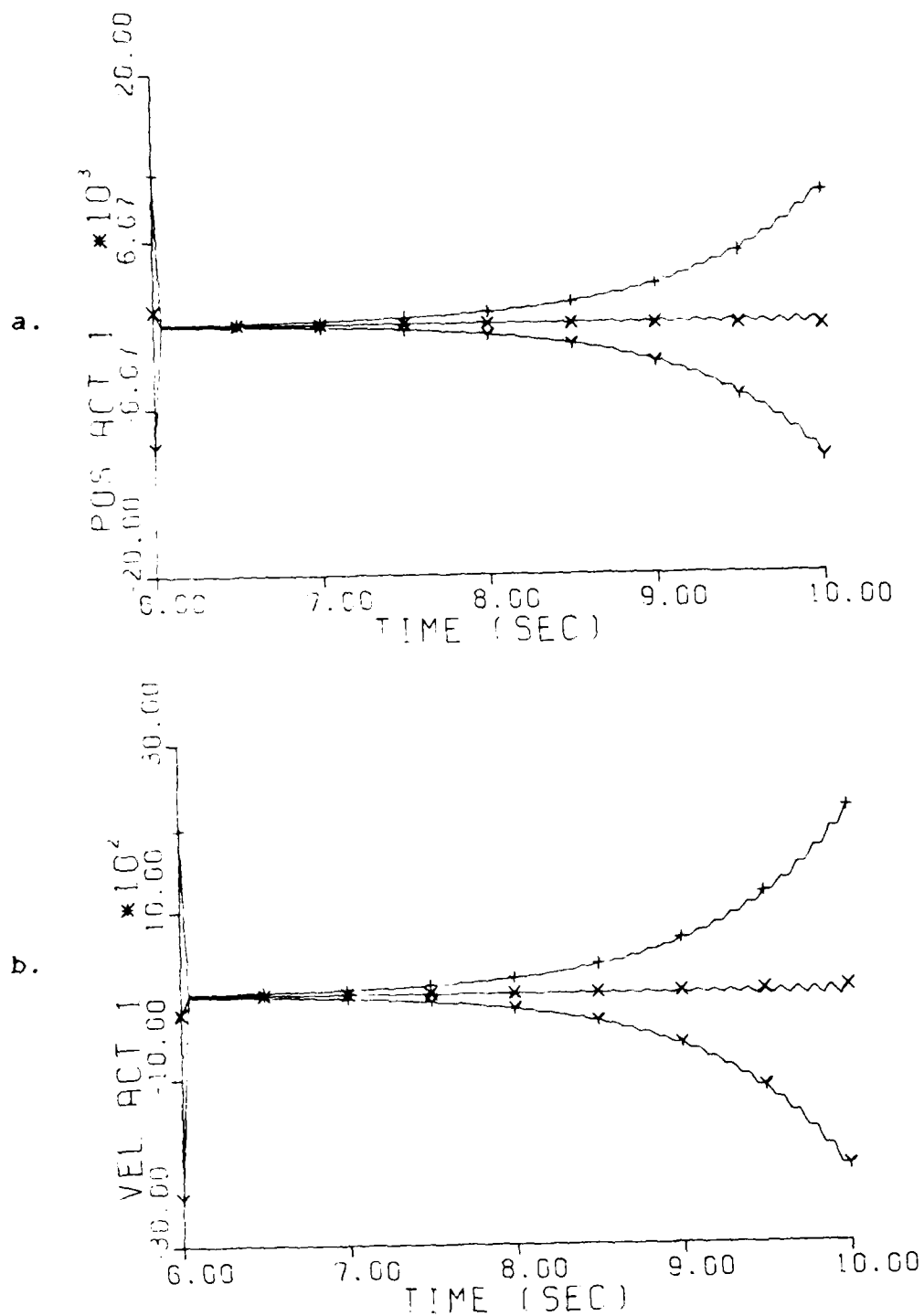


Fig. 5-31. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1\sigma$
 X and U Mistuned
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

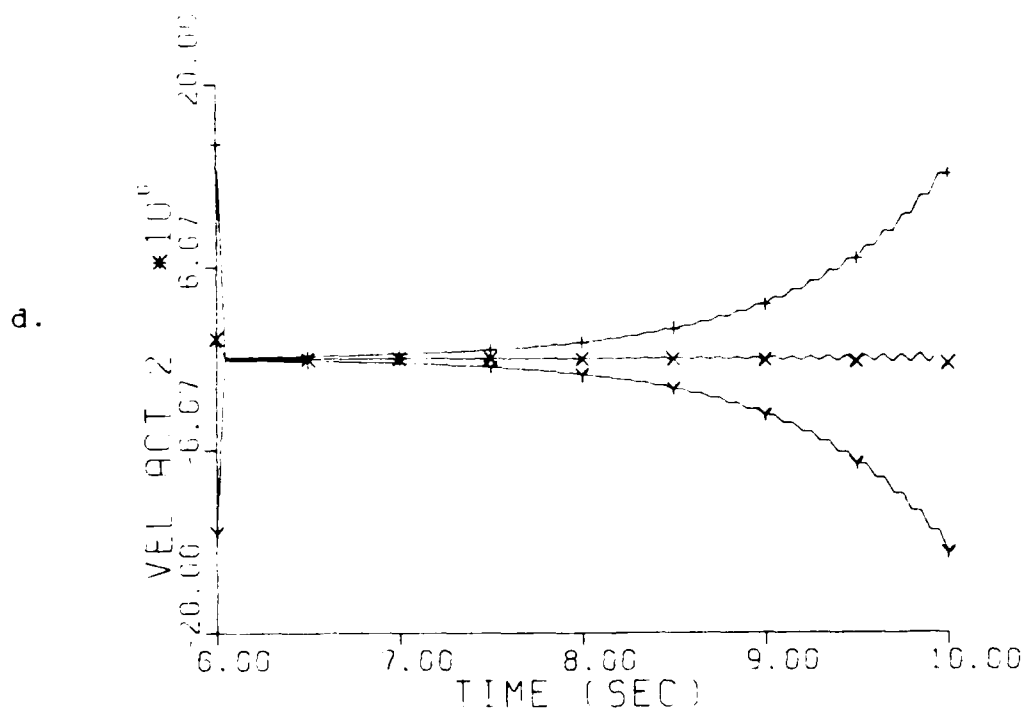
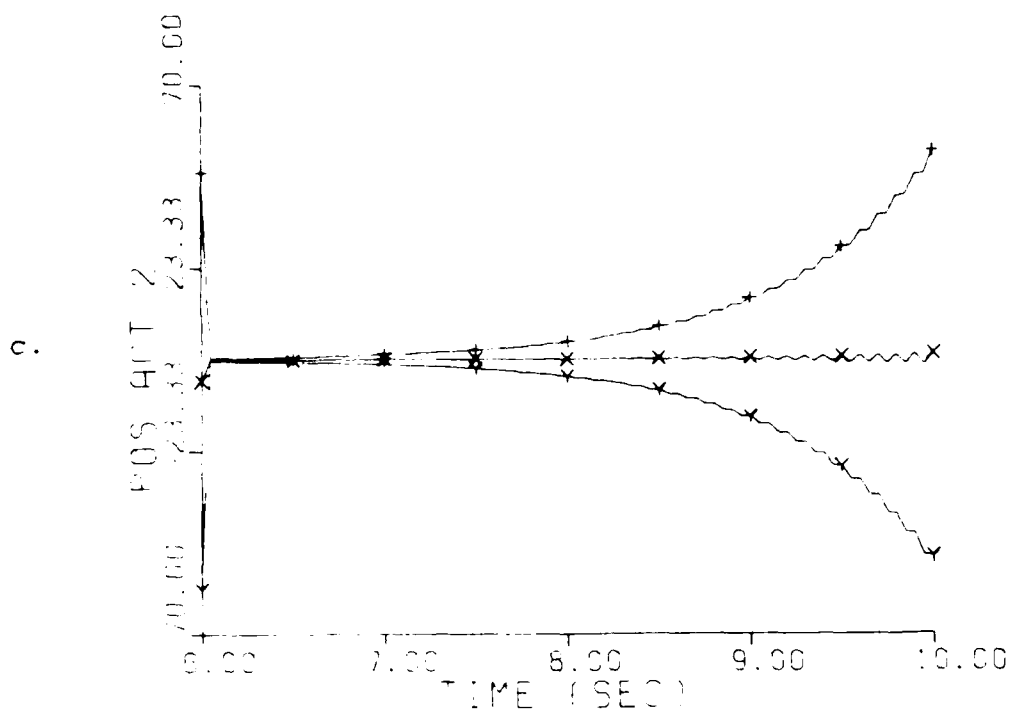


Fig. 5-31. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1\sigma$
 X and U Mistuned
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

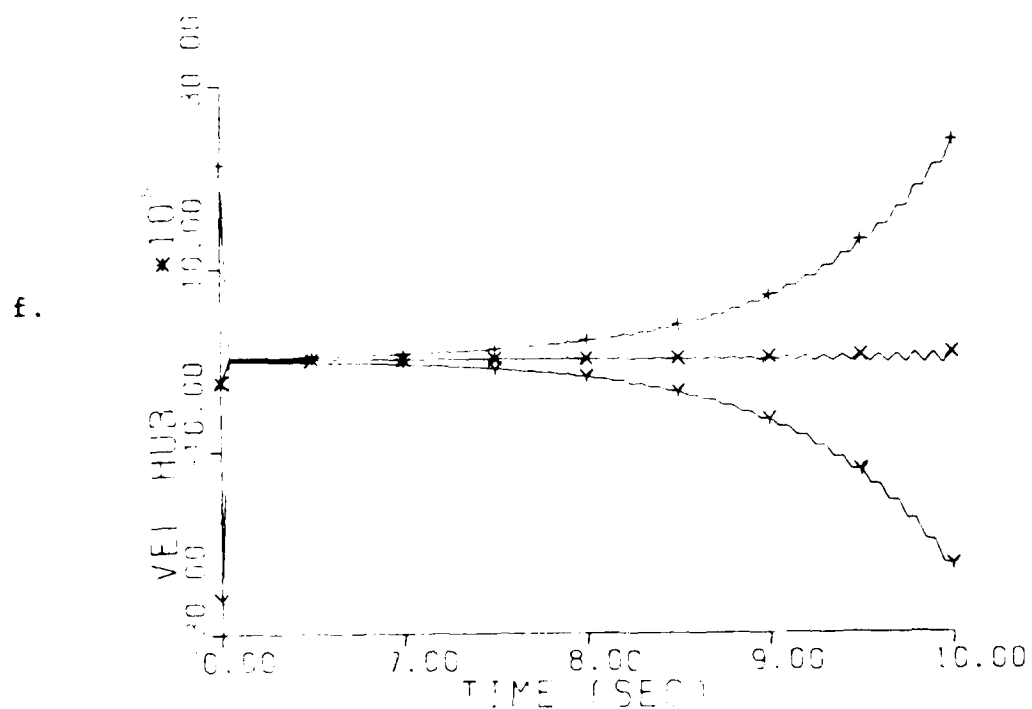
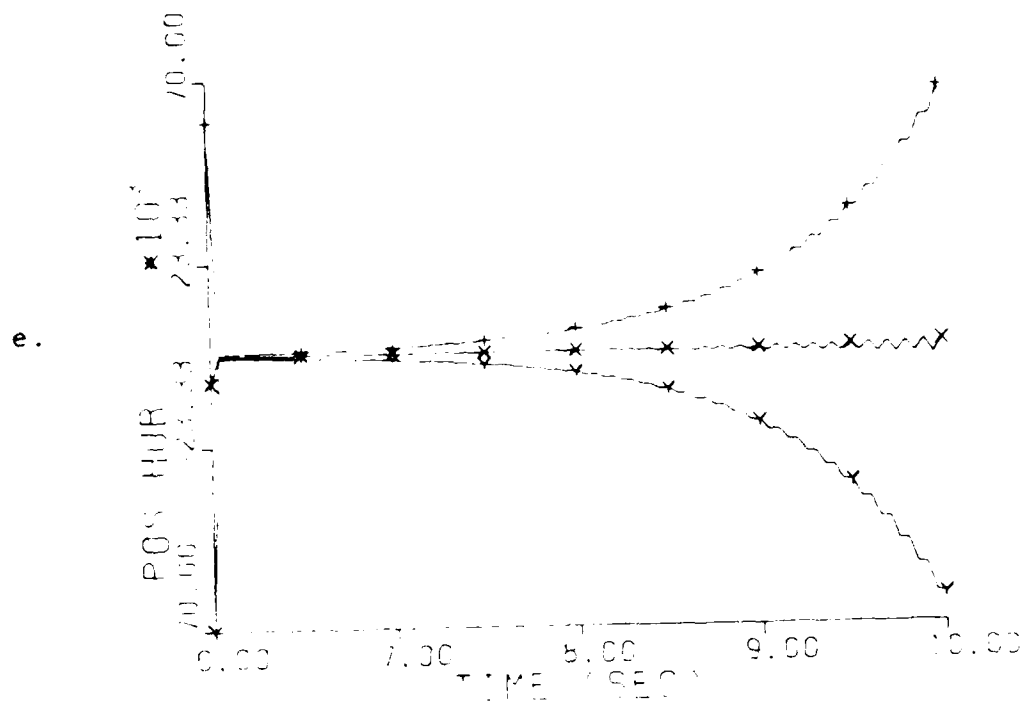


Fig. 5-31. Moving-Bank Controller; Post-Transient Performance

Parameter point (5,5); Mean $\pm 1 \sigma$
X and U Mistuned

e. Position State at node #3 (in)

f. Velocity State at node #3 (in/sec)

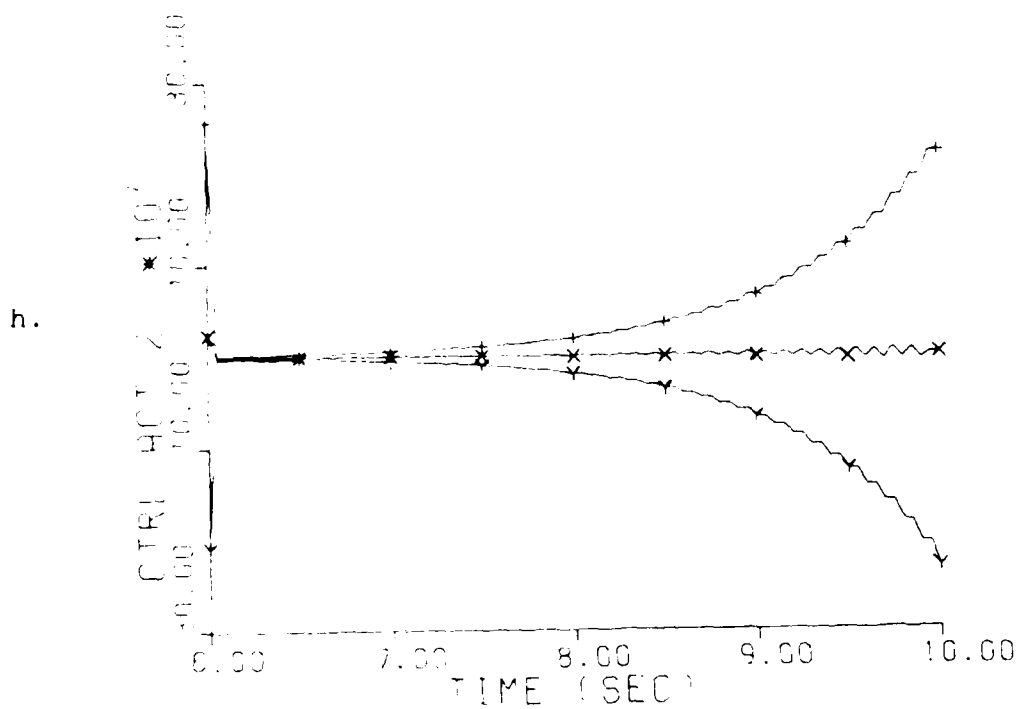
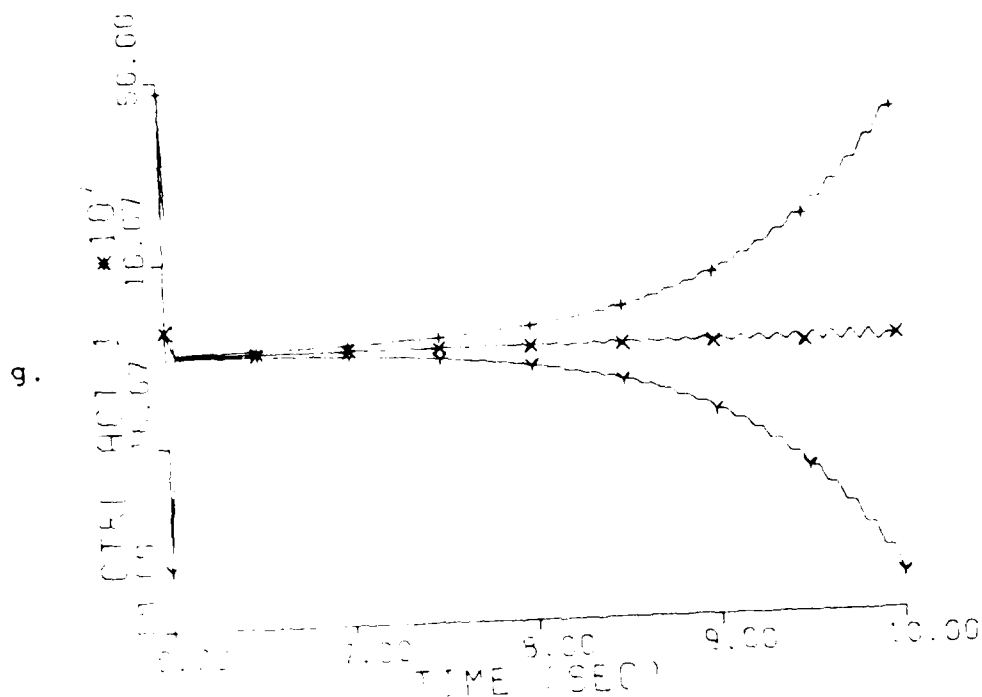


Fig. 5-31. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 10^\circ$
 X and U Mistuned
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

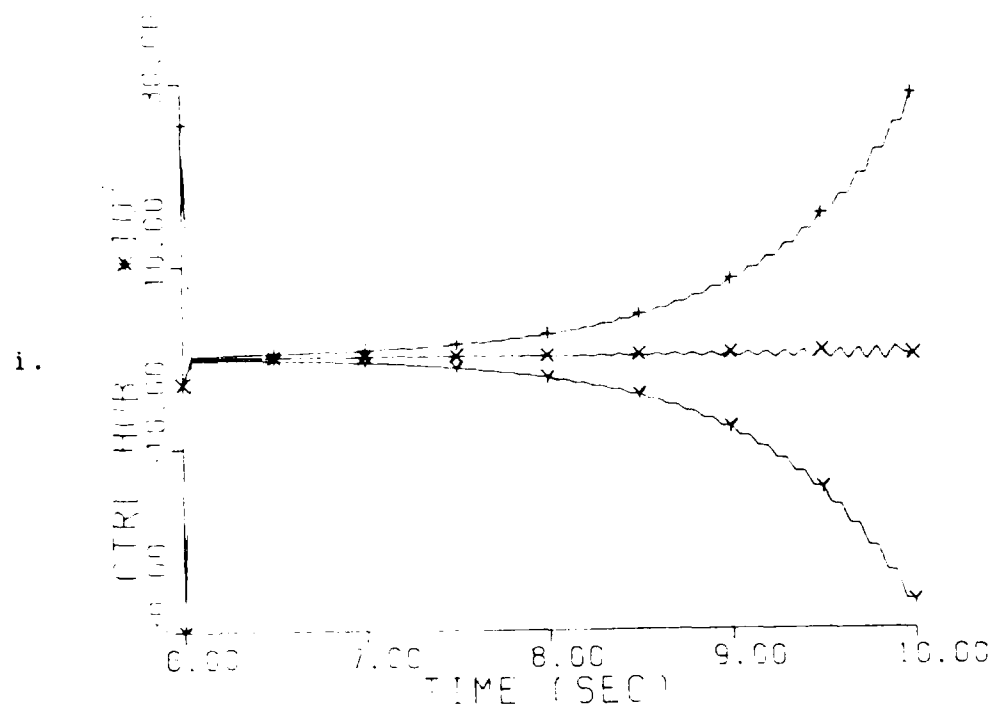


Fig. 5-31. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 i. Control Amplitude at node #3 (in)

generated. In both these figures, the X and U values obtained for the (7,6) parameter point (see Table 5-4) were used; however, the true parameter point was (5,5). As can be seen, both the moving-bank and fixed-bank controllers are unstable. Then the X and U values were retuned for the (5,5) parameter point (see Table 5-7). This time both the moving-bank and fixed-bank controllers provided a stable system (see Figures 5-32 and 5-33). Therefore, the X and U values must be tuned for each parameter point. This mistuning of certain parameter points could cause the poor performance of the fixed-bank controller for the (7,6) parameter point. However, the coarseness of discretization in the fixed-bank controller is still an issue: even with separate tuning for each implemented parameter point, there is still no single elemental filter/controller that is particularly well tuned for a parameter value far from any single coarsely discretized point value.

In addition to showing that each parameter point needs to have its X and U matrices tuned, Figures 5-32 and 5-34 also show that all three controllers are performing nearly identically. This is due to the fact that the true parameter happens to correspond to a parameter point in the fixed bank, then the fixed-bank algorithm will track the true parameter locks onto the actual parameter value. In fact, the fixed-bank algorithm will track the true parameter value.

AD-A190 761

MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED
TO FLEXIBLE SPACES. (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R M LASHLEE

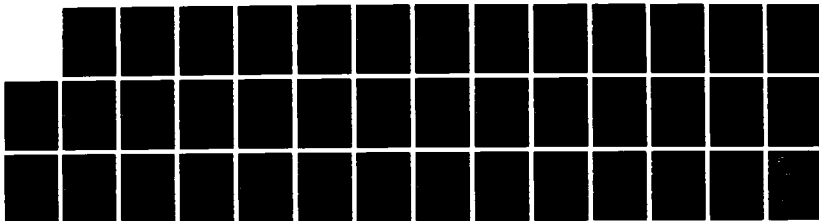
3/3

UNCLASSIFIED

DEC 87 AFIT/BE/ENG/87D-36

F/G 22/4

NL



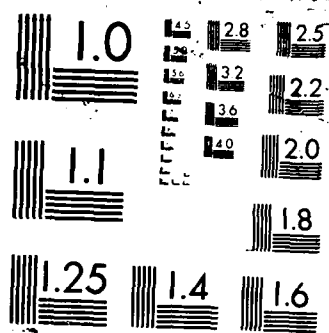


Table 5-7

State Weighting and Control Weighting Matrices
(for the (5,5) parameter point)
(Units: Position - in; Velocity - in/sec)

Position of Node 1 (X1) - 762,70.0

Position of Node 2 (X2) - 7917.8

Velocity of Node 1 (X3) - 508.5

Velocity of Node 2 (X4) - 82.0

Position of Hub (X5) - 10.0

Velocity of Hub (X6) - 1.0

Actuator at Node 1 (U1) - 3.0

Actuator at Node 2 (U2) - 3.0

Actuator at Node 3 (U3) - 9.0

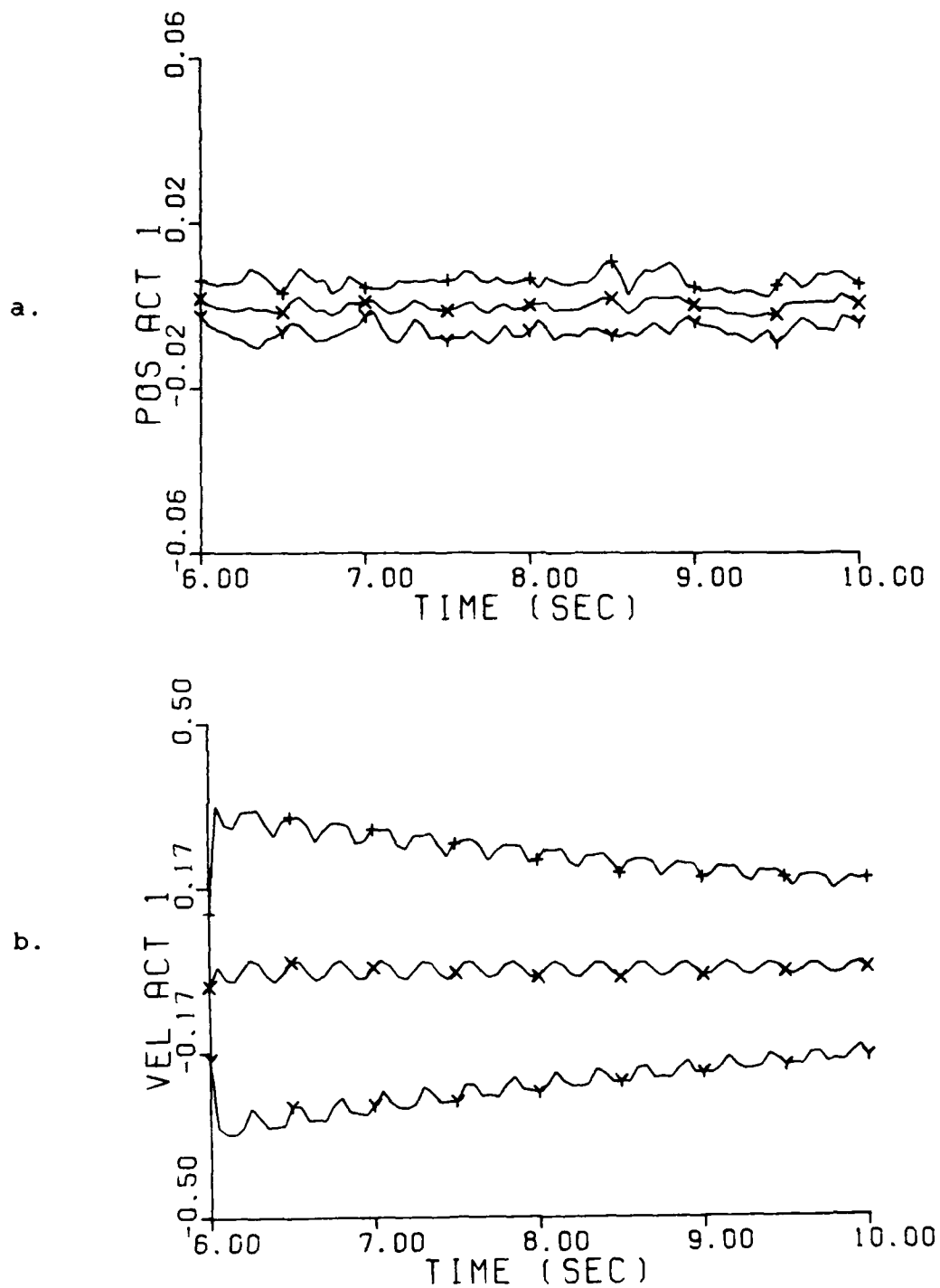


Fig. 5-32. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

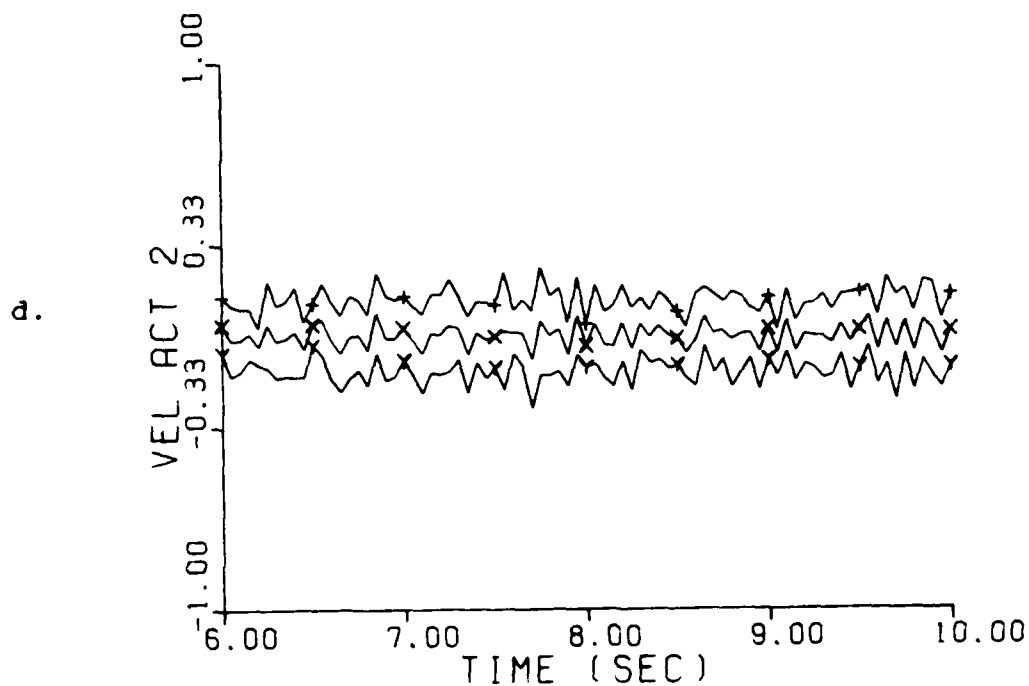
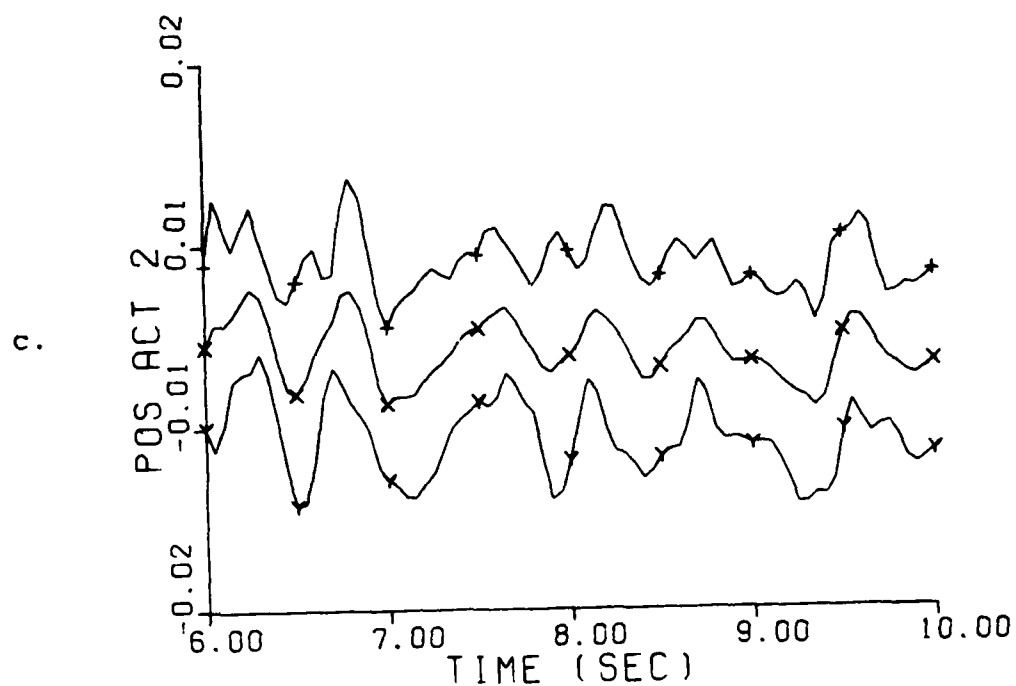


Fig. 5-32. Moving-Bank Controller; Post-Transient Performance

Parameter point (5,5); Mean $\pm 1 \sigma$
X and U Tuned

c. Position State at node #2 (in)

d. Velocity State at node #2 (in/sec)

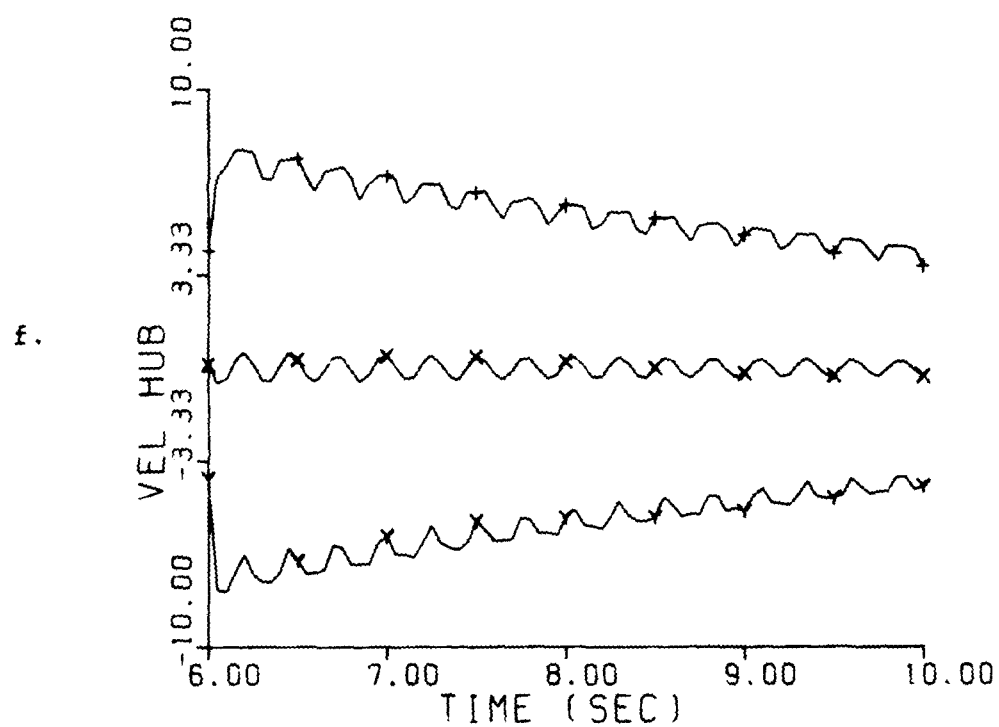
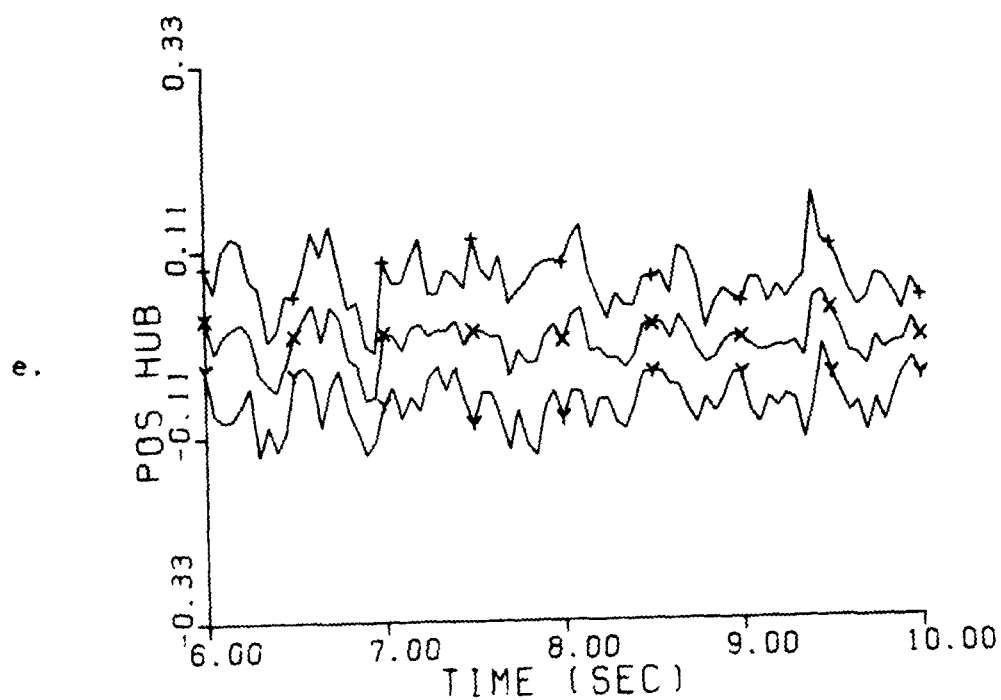


Fig. 5-32. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

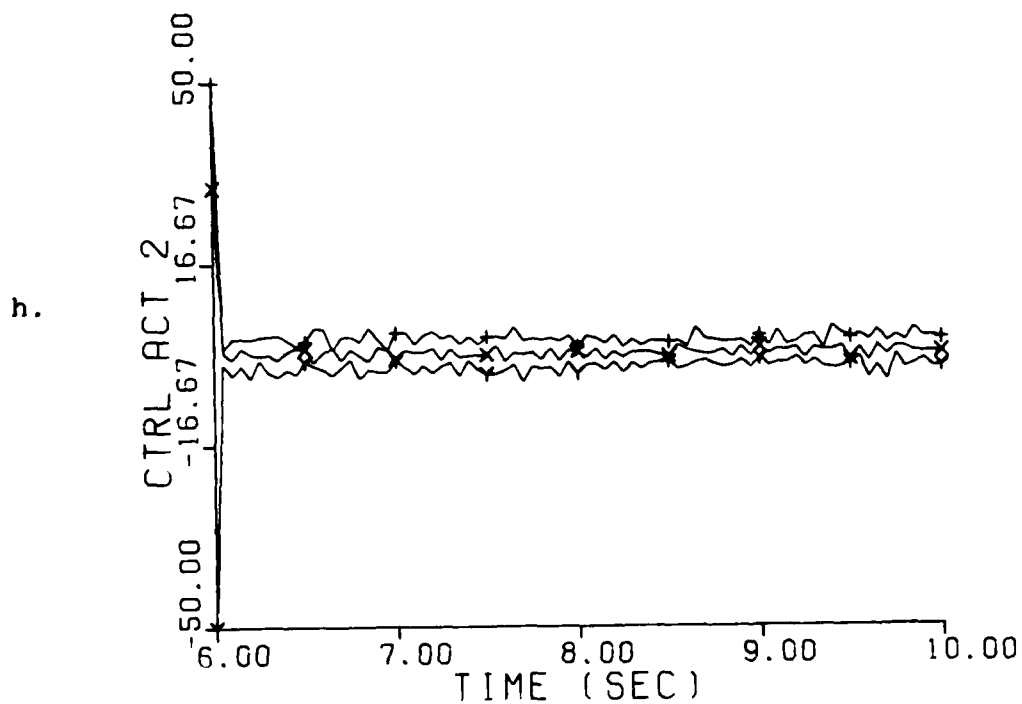
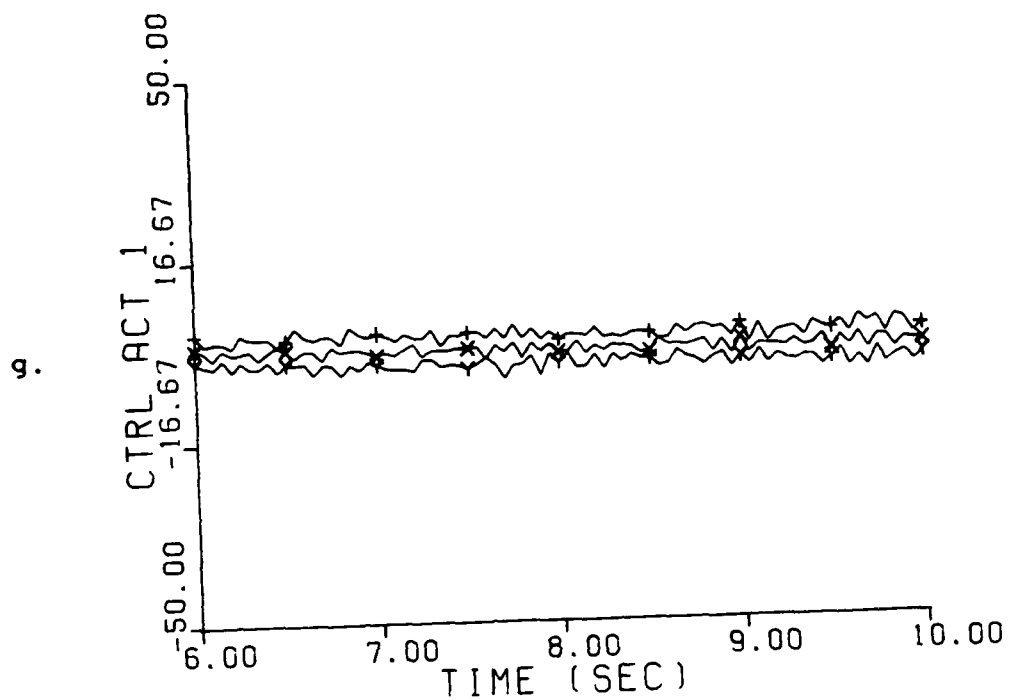


Fig. 5-32. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

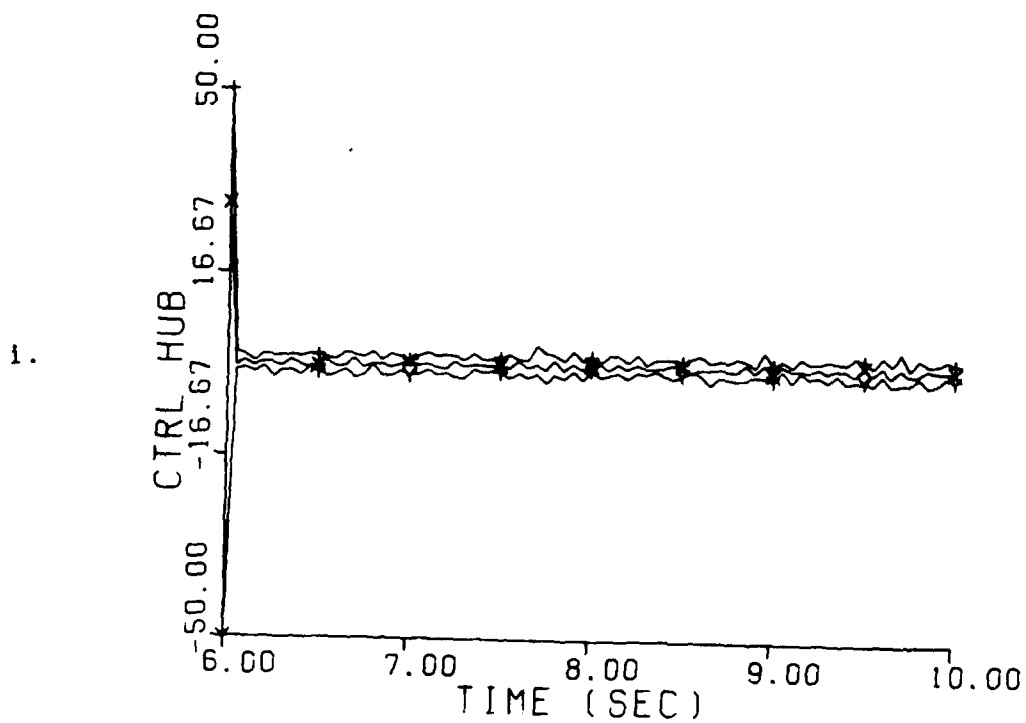


Fig. 5-32. Moving-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Mistuned
 1. Control Amplitude at node #3 (in)

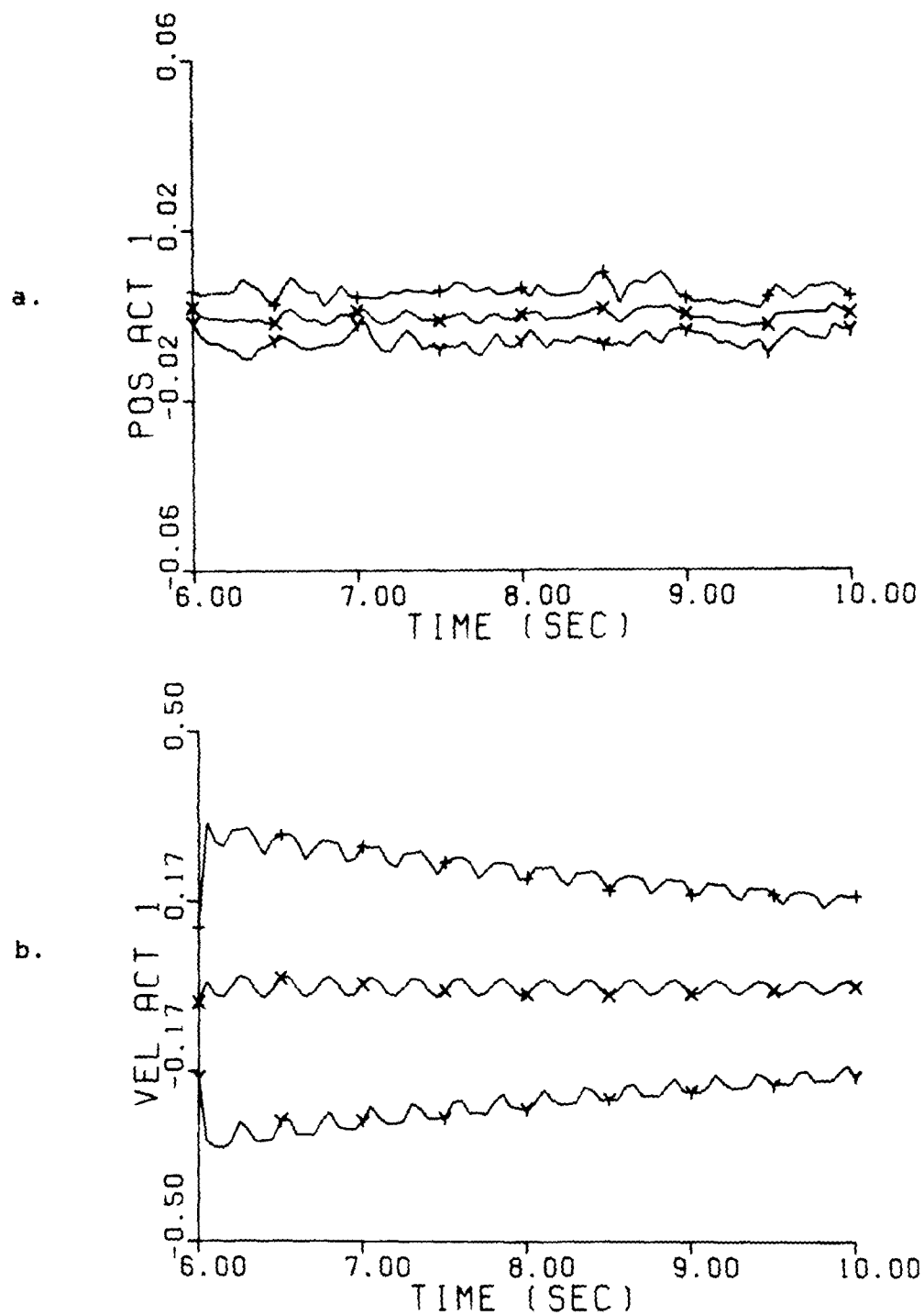


Fig. 5-33. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

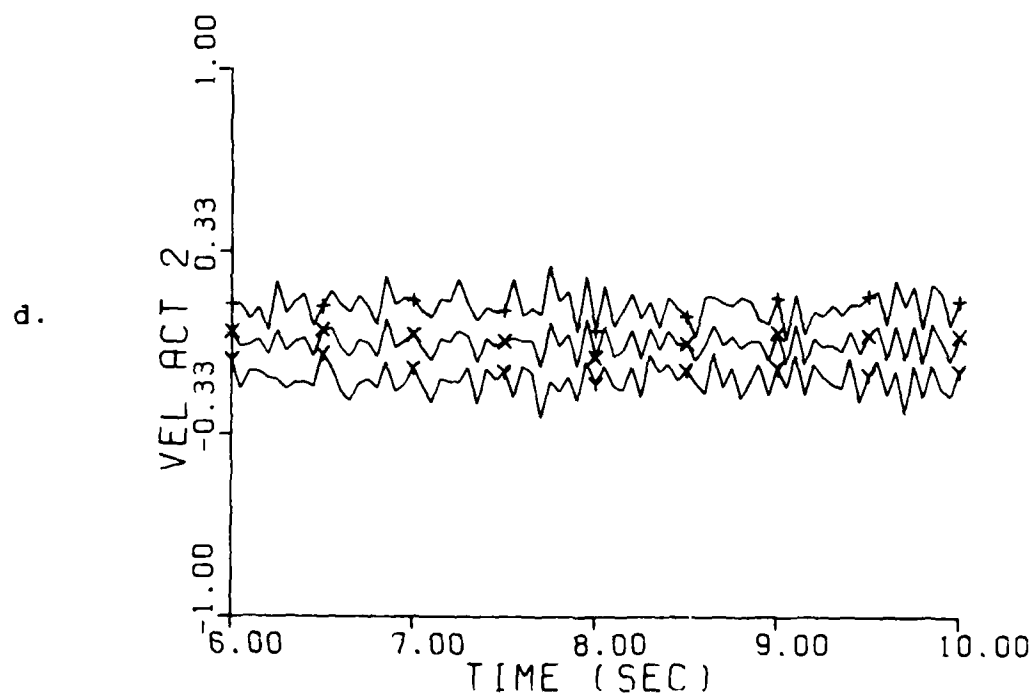
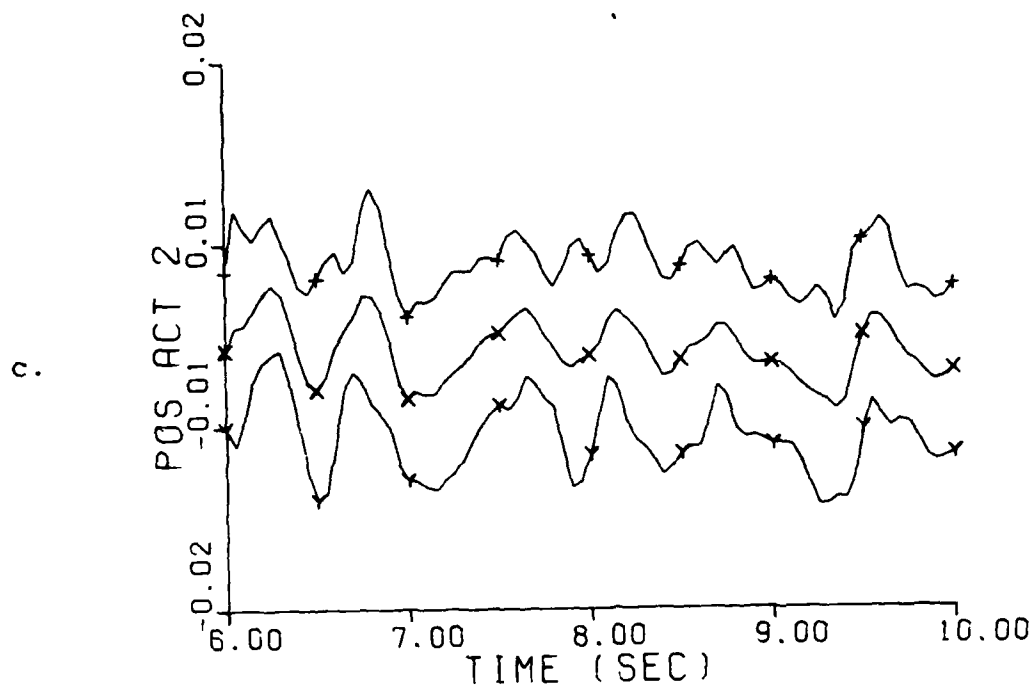


Fig. 5-33. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

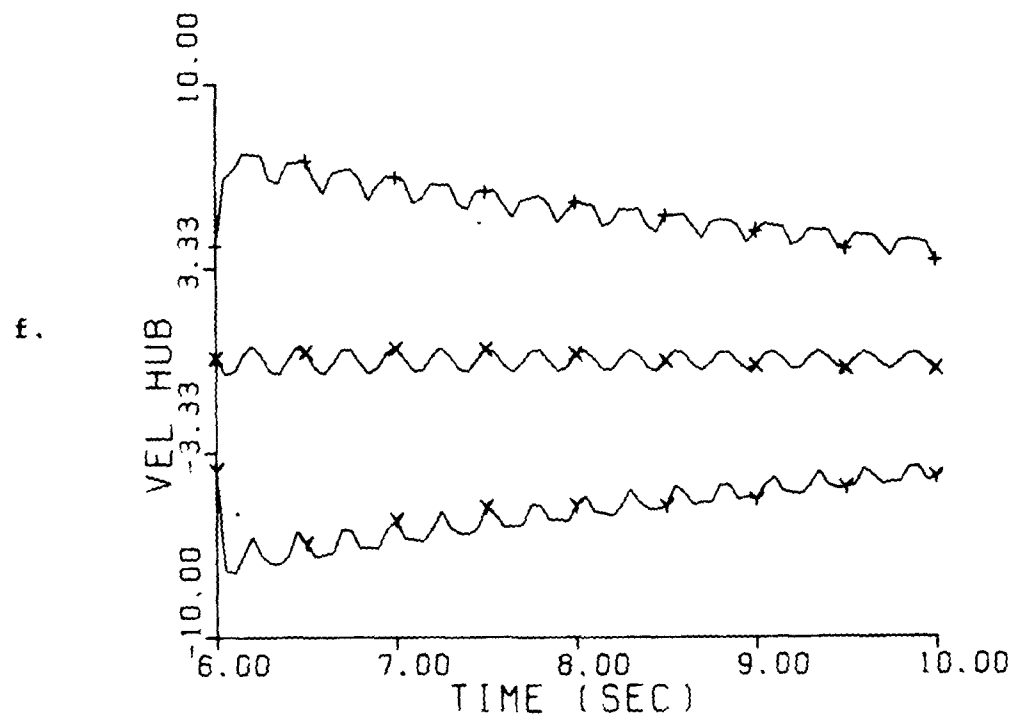
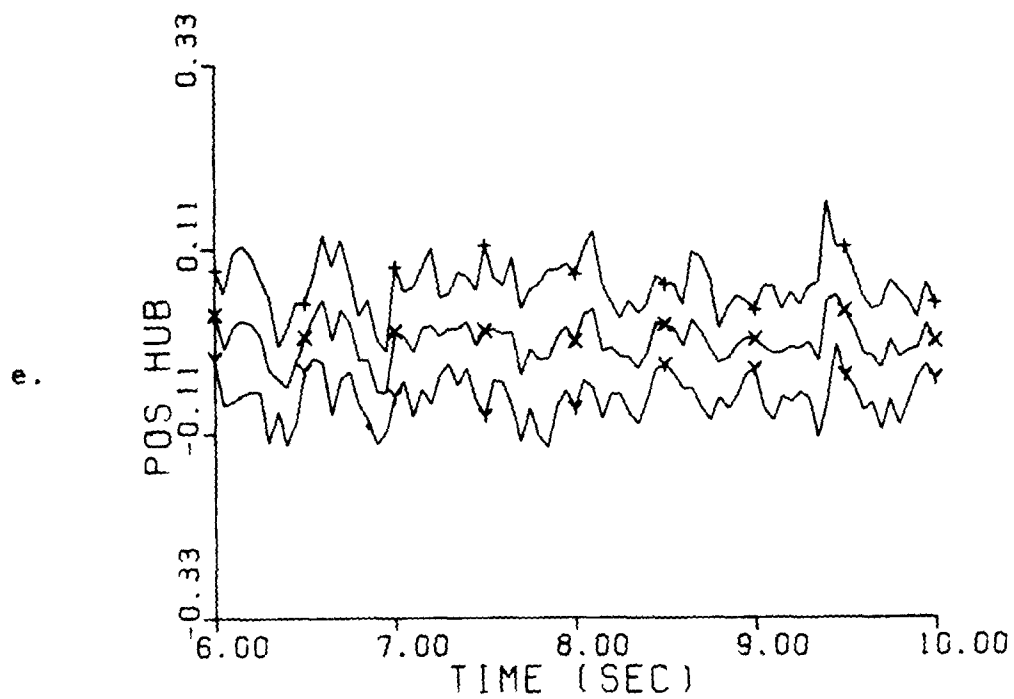


Fig. 5-33. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 e. Position State at node #3 (in)
 f. Velocity State at node #3 (in/sec)

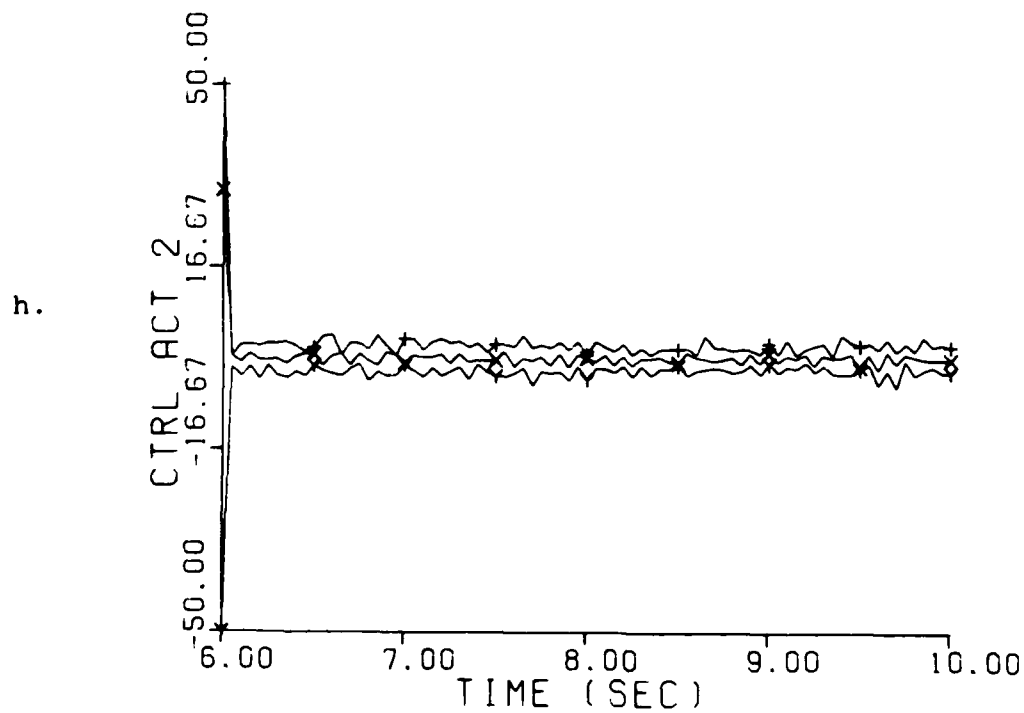
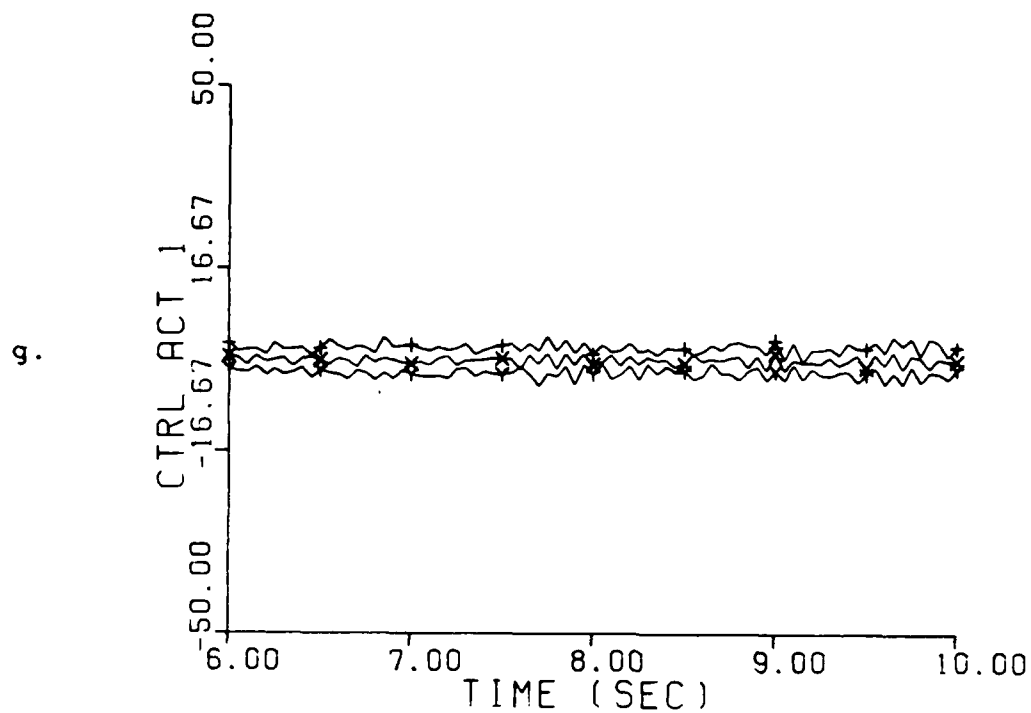


Fig. 5-33. Fixed-Bank Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

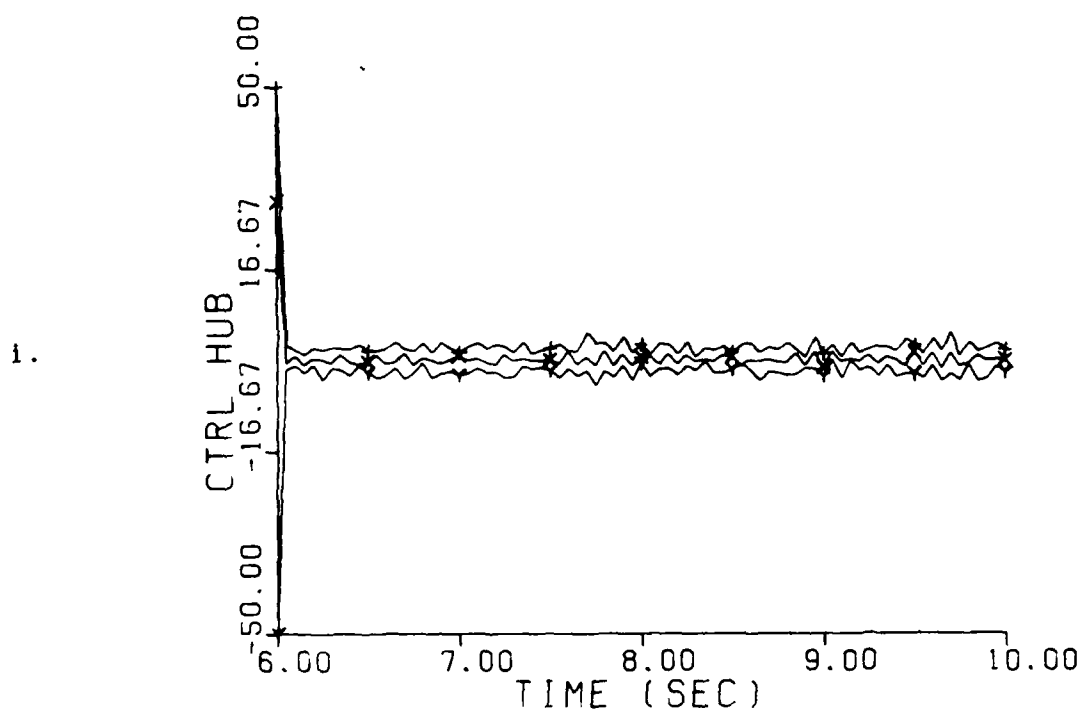


Fig. 5-33. Fixed-Bank Controller; Post-Transient
Performance
Parameter point (5,5); Mean $\pm 1 \sigma$
X and U Tuned
1. Control Amplitude at node #3 (in)

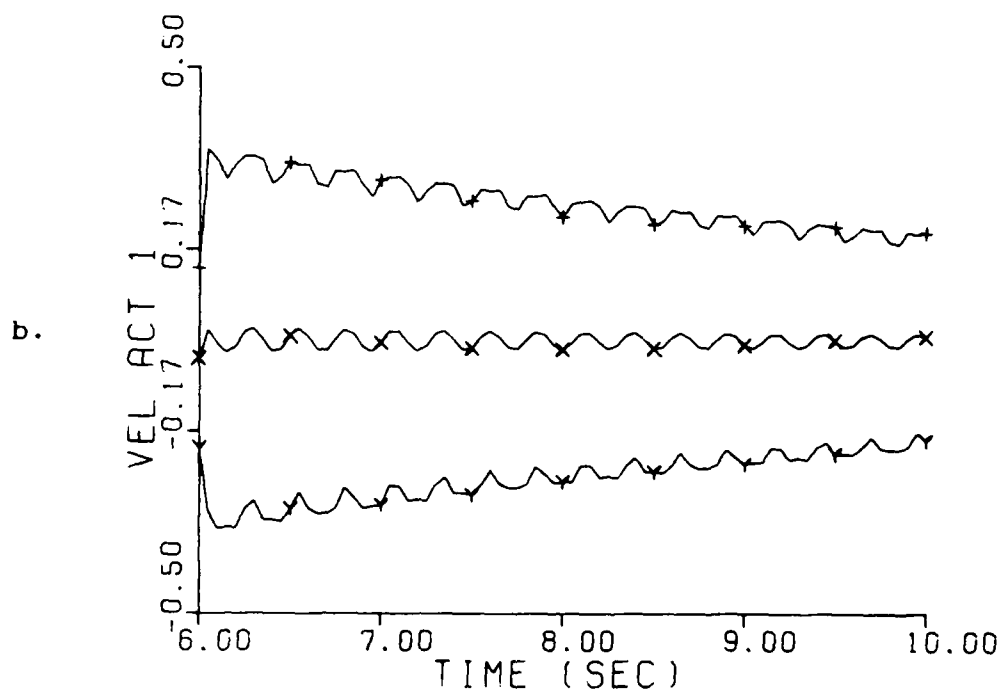
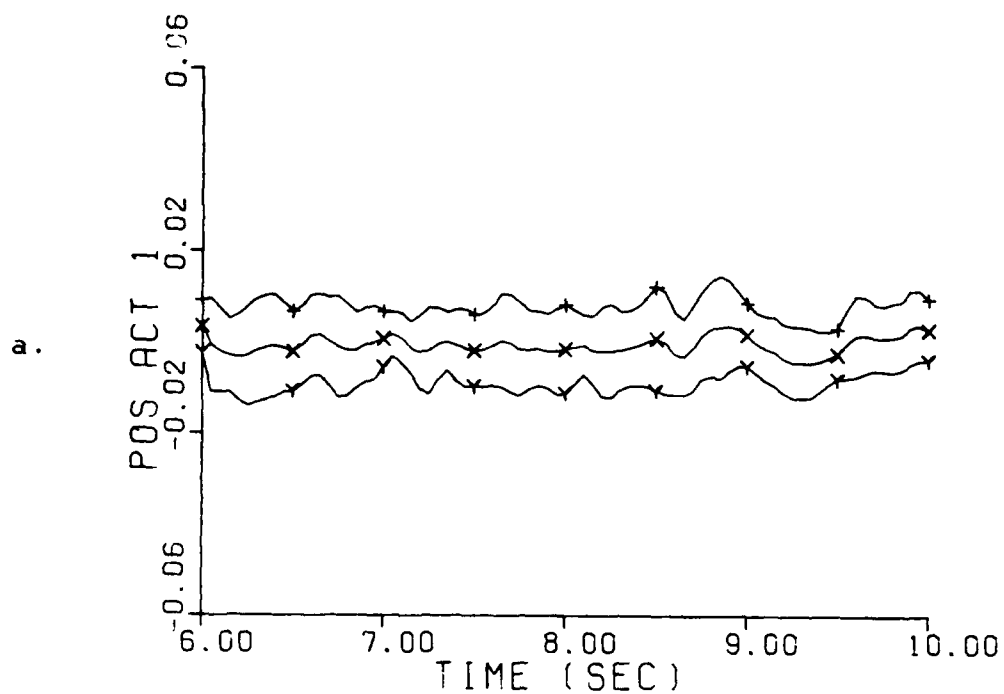


Fig. 5-34. Benchmark Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 a. Position State at node #1 (in)
 b. Velocity State at node #1 (in/sec)

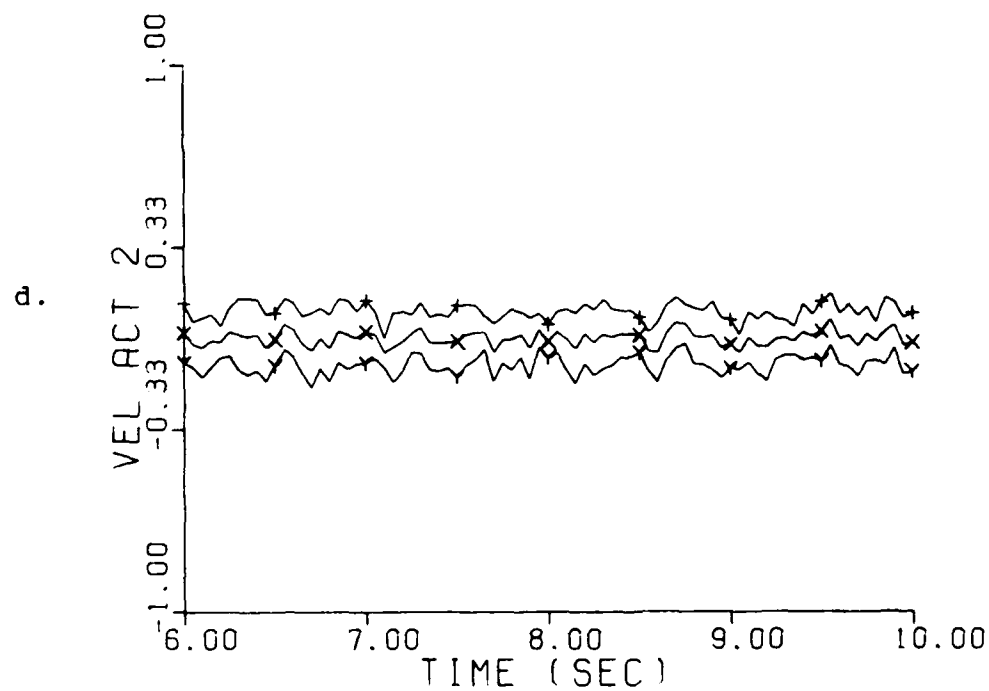
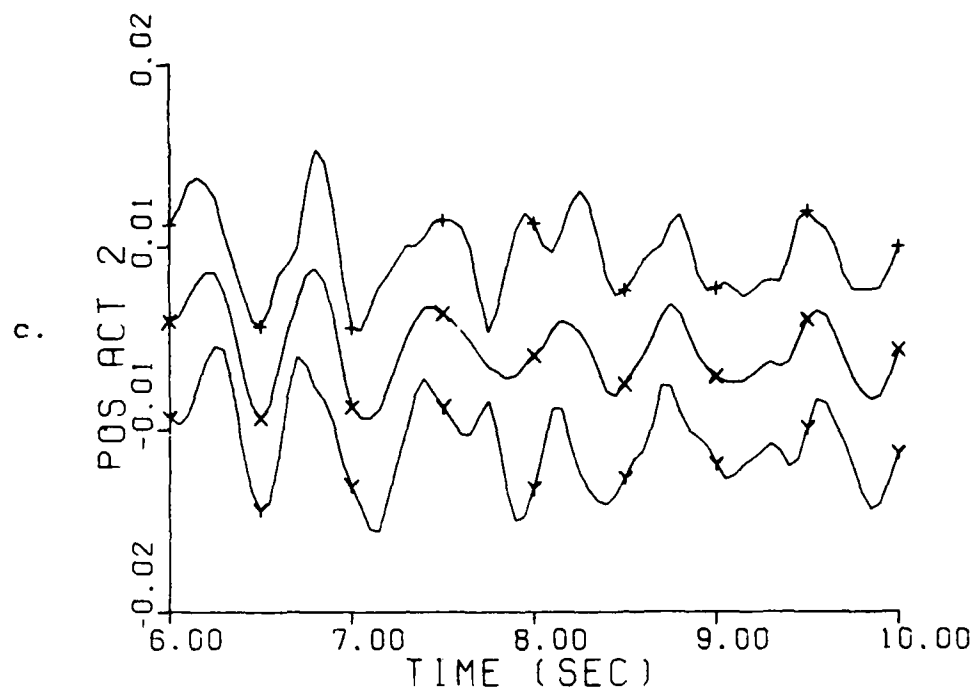


Fig. 5-34. Benchmark Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1 \sigma$
 X and U Tuned
 c. Position State at node #2 (in)
 d. Velocity State at node #2 (in/sec)

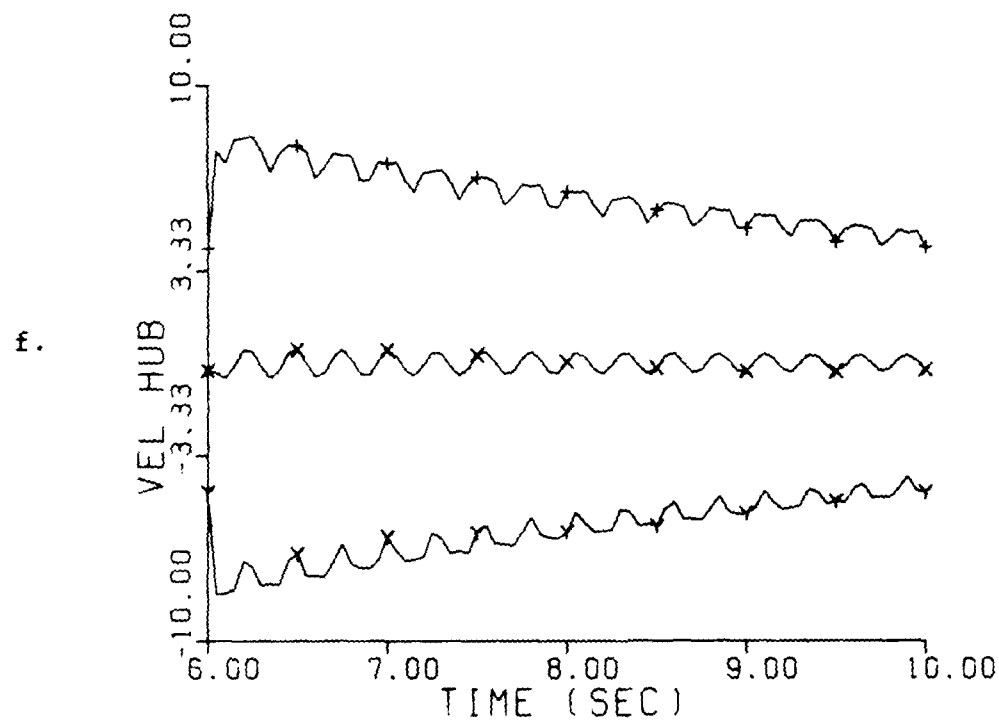
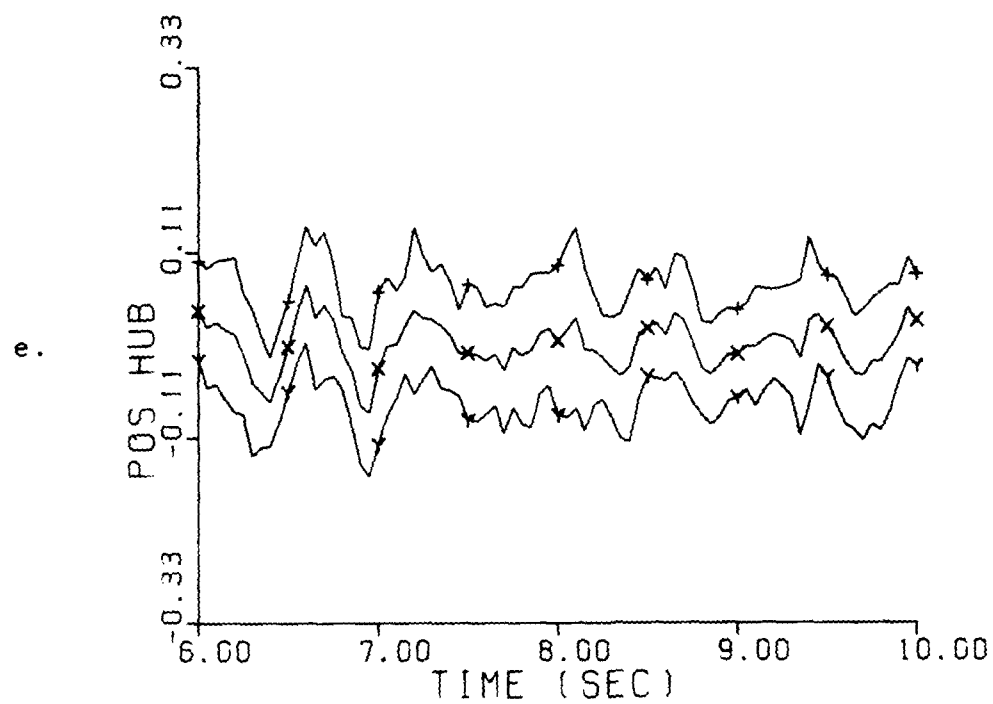


Fig. 5-34. Benchmark Controller; Post-Transient
Performance
Parameter point (5,5); Mean $\pm 1\sigma$
X and U Tuned
e. Position State at node #3 (in)
f. Velocity State at node #3 (in/sec)

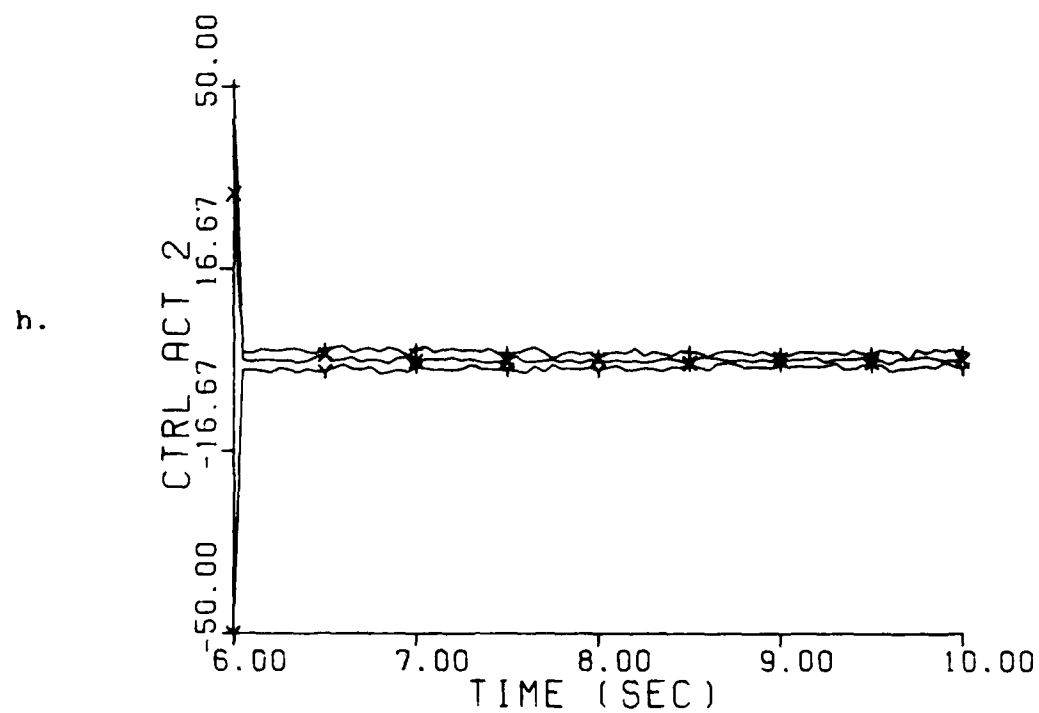
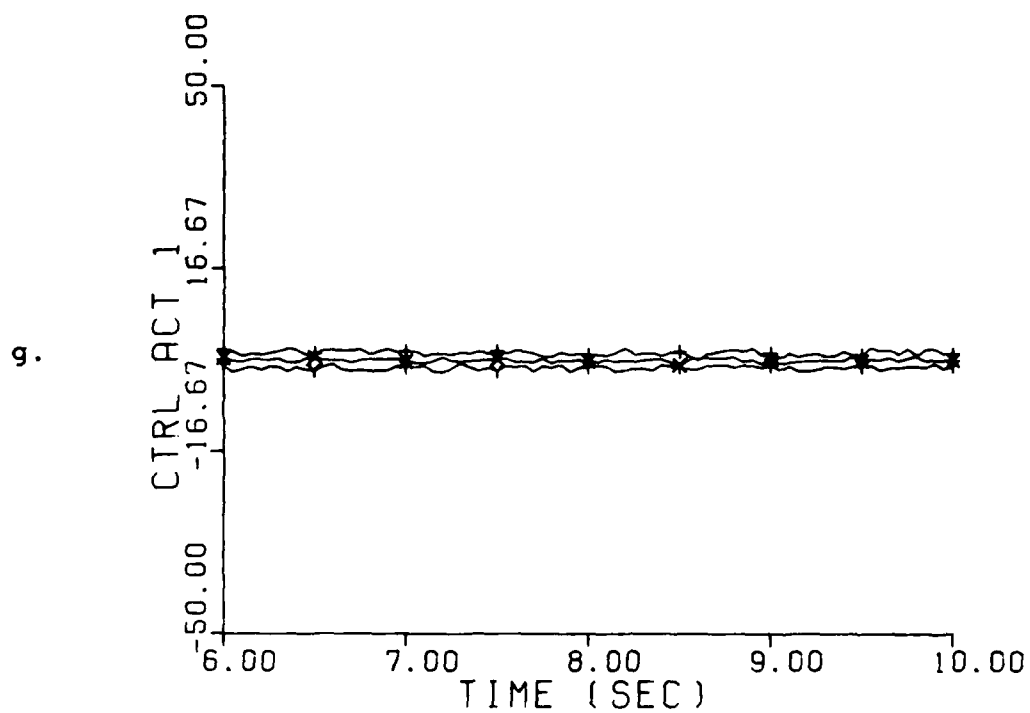


Fig. 5-34. Benchmark Controller; Post-Transient Performance
 Parameter point (5,5); Mean $\pm 1\sigma$
 X and U Tuned
 g. Control Amplitude at node #1 (in)
 h. Control Amplitude at node #2 (in)

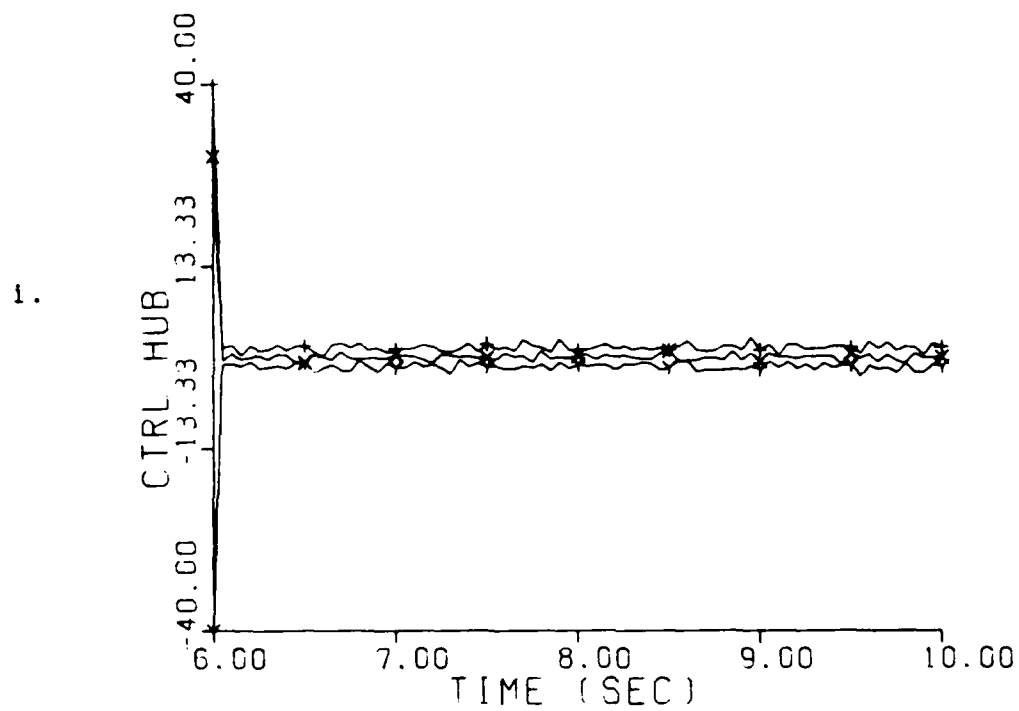


Fig. 5-34. Benchmark Controller; Post-Transient
Performance
Parameter point (5,5); Mean $\pm 1 \sigma$
X and U Tuned
1. Control Amplitude at node #3 (in)

moving-bank algorithm. Therefore, now both the fixed- and moving-bank controllers are converging to the benchmark controller performance. However, when the true parameter does not correspond to a filter in the fixed bank, then the fixed-bank algorithm's estimate of the true parameter jumps throughout the parameter space and its performance never converges to the performance of the benchmark controller, while the moving-bank controller characteristics are converging to those of the benchmark controller in both cases.

5.10. Summary

The results of this thesis effort were presented in this chapter. A number of important points were uncovered in this research. First, the problem with the moving-bank algorithm's performance in last year's effort by Karnick was due to the values of the measurement noise covariance and dynamics noise strength. Next, the space discretization study showed that the parameter space discretization plays a large role in the performance of the MMAE. Following the space discretization study, the cross term study showed that adding cross terms to the quadratic cost function did not substantially improve the MMAC performance. Finally, the moving-bank MMAC and the fixed-bank MMAC were compared with a benchmark controller. The results showed that the performance of the moving-bank

MMAC converged to that of the benchmark controller because of the speed at which the estimated true parameter converges on the actual true parameter value. In contrast, the fixed-bank algorithm performance converged to that of the benchmark controller if the true parameter happened to correspond to a filter in the fixed bank; however, if the true parameter did not correspond to a filter in the fixed bank, then the performance was substantially worse.

VI. Conclusions and Recommendations

6.1. Introduction

The investigation of the moving-bank Multiple Model Adaptive Estimator showed very encouraging performance in parameter estimation, state estimation, and control performance. The research performed in the effort should provide for a strong base for follow-on efforts.

6.2. Conclusions

The values of the measurement noise covariance (R) play an extremely large role in the performance of the moving-bank algorithm. In fact, this was the predominant problem with the moving-bank algorithm performance in last year's effort [9]. The important influence that R had on the moving bank performance was expected because large R values would tend to mask the differences between good and bad filters, thereby making it extremely difficult to discern anything distinctly about the system parameters. This limit on the R values is the major driver of the type of measurement devices used in a moving-bank MMAE problem.

Once the moving bank is being pulled in the right direction through proper choice of accurate enough sensors, additional performance can be achieved by determining the optimal space discretization level. The correct

discretization level can transform a poorly operating moving bank into an effectively operating moving bank. Again this is not surprising, because the easier it is for the bank to distinguish between good and bad filters, the better the moving-bank MMAE operates, both in state estimation and parameter estimation.

After the best moving-bank MMAE has been designed, the next step is to enhance the controller performance of the MMAC. For the large spacestructure model, the moving-bank MMAC needs to have the state weighting matrix and the control weighting matrix tuned for each parameter point. In addition to the tuning of the weighting matrices, the cross terms were shown to have little effect on the quadratic cost function for this problem's choice of sample period. Also, the fixed-bank MMAC performs as well as the moving-bank MMAC if the true parameter corresponds to one of the nine filters contained in the fixed-bank MMAC, but, the fixed-bank MMAC performance degrades substantially if the true parameter does not correspond to one of the filters in the fixed-bank. Thus, the moving-bank algorithm inherently has greater performance potential, since it does not require such a coarse discretization of parameter space in order to yield a practically implementable controller.

6.3. Recommendations

It is strongly recommended that research continue using the rotating two-bay truss model of a flexible space structure.

The most important recommendation by far is that the effect of unmodelled effects be investigated. This would be accomplished by keeping the filter dimensioned at six but increasing the "truth" dimension to 16 states or more. So far all the research has been conducted with equal order truth and filter models. This is not very realistic. Robustness of the parameter adaptation process, and ultimately of the state estimation and control performances, requires investigation.

It is also recommended that the state weighting matrix and control weighting matrix be tuned for each parameter point. This would entail minor modifications to the preprocessor and would allow the moving-bank MMAC to obtain better control performance.

In this study the true parameter value was always held constant, however the parameters can also be slowly varying or make jumps in parameter space. Therefore, a study needs to be conducted for varying true parameter values. The performance of the moving-bank algorithm could be monitored so that once the performance degrades to a predetermined point, then a reacquisition cycle can be

performed to reacquire a good estimate of the true parameter.

Finally, a performance comparison could be conducted between both the state estimates and the resulting controls produced with the Bayesian estimator and the state estimates and controls produced with a MAP estimator. Recall, the Bayesian estimation is formed by a probabilistically weighted-averaging of all the filters in the bank. On the other hand, the MAP estimate is formed by using the state estimate that corresponds to the filter with the highest probability in the bank. The performance comparison is based upon the accuracy of the state estimation and control produced by each method, not the accuracy of the parameter identification, because the main object is to design a good adaptive state estimator and controller, not a parameter identifier.

Appendix A: LQG Controller Development [9]

Assume the following stochastic system [3:20-22; 6:33-35]:

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{B}\underline{u}(t) + \underline{G}\underline{w}(t) \quad (\text{A-1})$$

where

$$\begin{aligned} E[\underline{w}(t)] &= \underline{0} \\ E[\underline{w}(t)\underline{w}^T(t+\tau)] &= \underline{Q} \delta(\tau) \end{aligned}$$

and the quadratic cost function to be minimized is:

$$J = E\left\{\int_0^\infty (1/2)[\underline{x}^T(t)\underline{W}_x\underline{x}(t) + \underline{u}^T(t)\underline{W}_u\underline{u}(t)]dt\right\} \quad (\text{A-2})$$

where \underline{W}_x and \underline{W}_u are weighting matrices to be chosen (iteratively) to yield a controller with desirable performance characteristics. The optimal constant-gain discrete linear feedback control law, assuming full-state access, is given by:

$$\underline{u}(t_i) = -\underline{G}_c^* \underline{x}(t_i) \quad (\text{A-3})$$

where the constant gains, \underline{G}_c^* , that minimize J , are given by: [23:68,122]

$$G_C^* = [U + B_d^T K_C B_d]^{-1} [B_d^T K_C \Phi + S^T] \quad (A-4)$$

where K_C satisfies the algebraic Riccati equation

$$K_C = X + \Phi^T K_C \Phi - [B_d^T K_C \Phi + S^T]^T G_C^* \quad (A-5)$$

and

$$X = \int_{t_i}^{t_{i+1}} \Phi^T(\tau, t_i) W_X \Phi(\tau, t_i) d\tau$$

$$U = \int_{t_i}^{t_{i+1}} [\bar{B}^T(\tau, t_i) W_X \bar{B}(\tau, t_i) + W_U] d\tau$$

$$S = \int_{t_i}^{t_{i+1}} \Phi^T(\tau, t_i) W_X \bar{B}(\tau, t_i) d\tau$$

$$\bar{B}(t, t_i) \triangleq \int_{t_i}^t \Phi(t, \tau) B d\tau$$

$$B_d = \bar{B}(t_{i+1}, t_i)$$

$\Phi(t_2, t_1)$ is the state transition matrix from t_1 to t_2
and the Φ 's in Equations (A-4) and (A-5) are:

$$\Phi = \Phi(t_{i+1}, t_i)$$

It should be noted that Equation (A-3) is also the solution for the deterministic LQ optimal control problem

with no driving noise $\underline{w}(t)$. If full state access is replaced by noise-corrupted measurements, $\underline{x}(t_i)$ in Equation (A-3) is replaced by the state estimate $\hat{\underline{x}}(t_i^+)$, which is often described as having the "certainty equivalence" property [21:17].

Appendix B: Rotating Two-Bay Truss System Matrices

This appendix lists the system matrices for the rotating two-bay truss. The reduced order matrices are developed from the mass and stiffness matrices (see Sections 3.2 and 3.4). The system is in modal formulation (see Section 3.3) and is composed of 6 states with 6 measurements and 4 control inputs. The six states were position at hub, position at actuator 1, position at actuator 2, velocity at hub, velocity at actuator 1, and velocity at actuator 2 respectively. The six measurements were position at actuator 1, position at actuator 2, velocity at actuator 1, velocity at actuator 2, position at hub, and velocity at hub respectively.

Appendix B
System Matrices

Stiffness Matrix

ROW 1	1.188E+3	1.966E+2	.000E+0	.000E+0
	-6.424E+2	.000E+0	-5.461E+2	-1.966E+2
	.000E+0	.000E+0	.000E+0	.000E+0
ROW 2	1.966E+2	6.263E+2	0.000E+0	-5.556E+2
	0.000E+0	0.000E+0	-1.966E+2	-7.077E+1
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 3	0.000E+0	0.000E+0	1.188E+3	-1.966E+2
	-5.461E+2	1.966E+2	-6.424E+2	0.000E+0
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 4	0.000E+0	-5.556E+2	-1.966E+2	6.263E+2
	1.966E+2	-7.077E+1	0.000E+0	0.000E+0
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 5	-6.424E+2	0.000E+0	-5.461E+2	1.966E+2
	4.019E+3	6.693E+1	0.000E+0	0.000E+0
	-2.099E+3	0.000E+0	-7.320E+2	-2.635E+2
Row 6	0.000E+0	0.000E+0	1.966E+2	-7.077E+1
	6.693E+1	7.212E+2	0.000E+0	-5.556E+2
	0.000E+0	0.000E+0	-2.635E+2	-9.487E+1
Row 7	-5.461E+2	-1.966E+2	-6.424E+2	0.000E+0
	0.000E+0	0.000E+0	4.019E+3	-6.693E+1
	-7.320E+2	2.635E+2	-2.099E+3	0.000E+0
Row 8	-1.966E+2	-7.077E+1	0.000E+0	0.000E+0
	0.000E+0	-5.556E+2	-6.693E+1	7.212E+2
	2.635E+2	-9.487E+1	0.000E+0	0.000E+0
Row 9	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	-2.099E+3	0.000E+0	-7.320E+2	2.635E+2
	8.618E+4	4.788E+4	0.000E+0	0.000E+0
Row 10	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	0.000E+0	0.000E+0	2.635E+2	-9.487E+1
	4.788E+4	1.390E+5	0.000E+0	-1.111E+5
Row 11	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	-7.320E+2	-2.635E+2	-2.099E+3	0.000E+0
	0.000E+0	0.000E+0	8.618E+4	-4.788E+4
Row 12	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	-2.635E+2	-9.487E+1	0.000E+0	0.000E+0
	0.000E+0	-1.111E+5	-4.788E+4	1.390E+5

Mass Matrix

Note that the first 8 diagonal elements are essentially the values of the non-structural mass because the non-structural mass is large compared to the structural mass.

Row 1	1.294E+0	-2.395E-6	0.000E+0	0.000E+0
	6.927E-6	0.000E+0	6.652E-6	2.395E-6
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 2	-2.395E-6	1.294E+0	0.000E+0	7.764E-7
	0.000E+0	0.000E+0	2.395E-6	8.621E-7
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 3	0.000E+0	0.000E+0	1.294E+0	2.395E-6
	6.652E-6	-2.395E-6	6.927E-6	0.000E+0
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 4	0.000E+0	7.764E-7	2.395E-6	1.294E+0
	-2.395E-6	8.621E-7	0.000E+0	0.000E+0
	0.000E+0	0.000E+0	0.000E+0	0.000E+0
Row 5	6.927E-6	0.000E+0	6.652E-6	-2.395E-6
	1.294E+0	-8.152E-7	0.000E+0	0.000E+0
	2.263E-5	0.000E+0	8.916E-6	3.210E-6
Row 6	0.000E+0	0.000E+0	-2.395E-6	8.621E-7
	-8.152E-7	1.294E+0	0.000E+0	7.764E-7
	0.000E+0	0.000E+0	3.210E-6	1.156E-6
Row 7	6.652E-6	2.395E-6	6.927E-6	0.000E+0
	0.000E+0	0.000E+0	1.294E+0	8.152E-7
	8.916E-6	-3.210E-6	2.263E-5	0.000E+0
Row 8	2.395E-6	8.621E-7	0.000E+0	0.000E+0
	0.000E+0	7.764E-7	8.152E-7	1.294E+0
	-3.210E-6	1.156E-6	0.000E+0	0.000E+0
Row 9	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	2.263E-5	0.000E+0	8.916E-6	-3.210E-6
	8.817E-4	-6.402E-5	0.000E+0	0.000E+0
Row 10	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	0.000E+0	0.000E+0	-3.210E-6	1.156E-6
	-6.402E-5	8.343E-4	0.000E+0	1.553E-4
Row 11	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	8.916E-6	3.210E-6	2.263E-5	0.000E+0
	0.000E+0	0.000E+0	8.817E-4	6.402E-5

Row 12	0.000E+0	0.000E+0	0.000E+0	0.000E+0
	3.210E-6	1.156E-6	0.000E+0	0.000E+0
	0.000E+0	1.553E-4	6.402E-5	8.343E-4

A Matrix

0.000E+0	0.000E+0	0.000E+0	1.000E+0	0.000E+0	0.000E+0
0.000E+0	0.000E+0	0.000E+0	0.000E+0	1.000E+0	0.000E+0
0.000E+0	0.000E+0	0.000E+0	0.000E+0	0.000E+0	1.000E+0
0.000E+0	0.000E+0	0.000E+0	0.000E+0	0.000E+0	0.000E+0
0.000E+0	-7.918E+1	0.000E+0	0.000E+0	-8.898E-2	0.000E+0
0.000E+0	0.000E+0	-5.085E+2	0.000E+0	0.000E+0	-2.255E-1

Rows 1 and 4 correspond to the rigid body angular position and velocity, respectively. Rows 2 and 3 correspond to the position of the first and second bending modes; respectively, while rows 5 and 6 represent the velocity of these bending modes, respectively.

B Matrix

0.000E+0	0.000E+0	0.000E+0	1.000E+0
0.000E+0	0.000E+0	0.000E+0	0.000E+0
0.000E+0	0.000E+0	0.000E+0	0.000E+0
-4.773E-1	-2.711E-1	1.000E+0	0.000E+0
2.589E-1	-4.736E-1	0.000E+0	0.000E+0
9.488E-2	1.590E-1	0.000E+0	0.000E+0

The first two columns represent inputs from actuators located on the truss while the third and fourth columns are due to an actuator located on the hub which induces inputs at both the position and velocity level (see Section 4.4.4.). The non-zero portion of the first two

columns was designed in physical coordinates in the unreduced system as

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

and then transformed into modal coordinates. The angular input actuator was then added by augmenting a column and entering a 1.00 in the row corresponding to the angular velocity state and the angular position state.

H Matrix

$$\begin{bmatrix} 6.066\text{E-}1 & -3.239\text{E-}1 & -1.228\text{E-}1 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 3.444\text{E-}1 & 5.925\text{E-}1 & -2.057\text{E-}1 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 6.066\text{E-}1 & -3.239\text{E-}1 & -1.228\text{E-}1 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 3.444\text{E-}1 & 5.925\text{E-}1 & -2.057\text{E-}1 \\ 1.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 1.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \end{bmatrix}$$

The first and second rows represent measurements from the position sensors located at node 1 and node 2. The third and fourth rows represent measurements from velocity sensors located at node 1 and node 2. Finally, the last two rows represent measurements of the angular position and velocity of the hub (see Section 3.4.4.). The position and velocity portion of the matrix were calculated in physical coordinates and then transformed into modal coordinates (see Section 3.3.), where the angular

measurements were augmented. Note that the entries in rows 1 and 2 are identical to those in rows 3 and 4 because of co-location of position and velocity sensors. Rows 1 and 2 were designed in physical coordinates in the unreduced system as:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Only the first twelve columns are listed because the remaining columns are zero.

D_r Matrix

$$\begin{bmatrix} -4.253\text{E-}4 & 6.396\text{E-}5 & 0.000\text{E-}0 & 0.000\text{E+}0 \\ 6.396\text{E-}5 & -3.583\text{E-}4 & 0.000\text{E-}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \\ 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 & 0.000\text{E+}0 \end{bmatrix}$$

The D_r matrix (see Section 3.5) is used in a method of order reduction referred to as singular perturbations. This thesis used reduced order models; however, this method was not implemented because truth models were of the same order.

Bibliography

1. Chang, C.B. and M. Athans. "State Estimation for Discrete Systems with Switching Parameters," IEEE Transactions on Aerospace and Electronic Systems, 14 (3): 418-424 (May 1978).
2. Dasgupta, S. and L. C. Westphal. "Convergence of Partitioned Adaptive Filters for Systems with Unknown Biases," IEEE Transactions on Automatic Control, 28: 614-615 (May 1983).
3. Filios, Paul G. Moving-Bank Multiple Model Adaptive Algorithms Applied to Flexible Spacecraft Control. MS thesis, GE/ENG/85D-14. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984.
4. Fry, C. M. and A. P. Sage. "On Hierarchical Structure Adaption and Systems Identification," International Journal of Control, 20: 433-452 (1979).
5. Hawkes, Richard M. and John B. Moore. "Performance Bounds for Adaptive Estimation," Proceedings of the IEEE, 64: 1143-1150 (August 1976).
6. Hentz, K. P. Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms. MS thesis, GE/ENG/84D-32. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984.
7. Hilborn, C. G., Jr. and Demetrios G. Lainiotis. "Optimal Estimation in the Presence of Unknown Parameters," IEEE Transactions on Systems Science and Cybernetics, 5: 38-43 (January 1969).
8. Kailath, Thomas. Linear Systems. Englewood Cliffs NJ: Prentice-Hall, Inc., 1980.

9. Karnick, D. Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control. MS thesis, GE/ENG/86D-41. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.
10. Kokotovic, P. V., R. E. O'Malley, Jr., and P. Sannuti. "Singular Perturbations and Order Reduction in Control Theory--An Overview," Automatica, 12: 123-132 (1976).
11. Kokotovic, P. V. and R. A. Yackel. "Singular Perturbation of Linear Regulators: Basic Theorems," IEEE Transactions on Automatic Control, 17: 29-37 (1972).
12. Korn, J. and L. Beean. Application of Multiple Model Adaptive Estimation to Maneuver Detection and Estimation. Contract DAAK10-82-C-0020. Alphatech, Inc., Burlington MA, June 1981DDDD (AD-B075 921).
13. Lainiotis, Demetrios G. "Optimal Adaptive Estimation: Structure and Parameter Adaptation," IEEE Transactions on Automatic Control, 16: 160-170 (April 1971).
14. Lainiotis, Demetrios G. "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation," Proceeding of the IEEE, 64: 1126-1143 (August 1976).
15. Lamb, P. R. and L. C. Westphal. "Simplex-Directed Partitioned Adaptive Filters," International Journal of Control, 30: 617-627 (1979).
16. Loving, P. A. Bayesian vs MAP Multiple Model Adaptive Estimation for Field of View Expansion in Tracking Airborne Targets. MS thesis, GE/ENG/85M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1985 (AD-A155 466).

17. Lynch, P. J. and Silva S. Banda. "Active Control for Vibration Damping," Damping: 1986 Proceedings, Technical Report. Flight Dynamics Laboratory AFWAS/FIGC, Wright-Patterson AFB OH, 1986 (AFWAL-TR-86-3509).
18. Magill, D. T. "Optimal Adaptive Estimation of Sample Stochastic Processes," IEEE Transactions on Automatic Control, 10(5): 434-439 (October 1965).
19. Maybeck, P. S. and K. P. Hentz. "Investigation of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms," Proceedings of the 24th IEEE Conference on Decision and Control, Ft. Lauderdale FL, December 1985.
20. Maybeck, P. S. and R. I. Suizu. "Adaptive Field of View Expansion Via Multiple Model Filtering for Tracking Dynamic Target Images," Proceedings of the IEEE 1985 National Aerospace and Electronic Conference. Dayton OH, May 1985.
21. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 1. New York: Academic Press, 1979.
22. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 2. New York: Academic Press, 1982.
23. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 3. New York: Academic Press, 1983.
24. Maybeck, Peter S. Discussions concerning Moving-Bank Multiple Model Adaptive Estimator, March-October 1987.
25. Meer, D. E. Multiple Model Adaptive Estimation for Space-Time Point Process Observations. PhD dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1982.
26. Meer, D. E. and P. S. Maybeck. "Multiple Model Adaptive Estimation for Space-Time Point Process Observations," Proceedings of the IEEE Conference on Decision and Control, Las Vegas NV, December 1984.

27. Meirovitch, Leonard. Analytical Methods in Vibrations. New York: The Macmillan Company, 1967.
28. Moose, R. L. and P. P. Wang. "An Adaptive Estimator with Learning for a Plant Containing Semi-Markov Switching Parameters," IEEE Transactions on Systems Science and Cybernetics, 277-281, May 1973.
29. Netzer, Alan S. Characteristics of Bayesian Multiple Model Adaptive Estimation for Tracking Airbone Targets. MS thesis, gae/eng/85D-2 School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985 (AD-A163 830).
30. Tang, Wang, and Gregory L. Mealy. "Application of Multiple Model Estimation Techniques to a Recursive Terrain Height Correlation System," IEEE Transactions on Automatica Control, 28: 315-323 (March 1983).
31. Turn, J. D. and H. M. Chun. "Optimal Distributed Control of a Flexible Spacecraft During a Large-Angle Rotational Maneuver," Proceedings of the Third VPI&SU/AIAA Symposium, 471-485, Blacksburg VA, June 1981.
32. Venkayya, V. B. and V. A. Tischler. "Frequency Control and Its Effects on the Dynamic Response of Flexible Structures," AIAA Journal, 23(11): 1768-1774 (November 1985).
33. Venkayya, V. B. and V. A. Tischler. 'Analyze'--Analysis of Aerospace Structures with Membrane Elements. Technical Report. Analysis and Optimization Branch, Structural Mechanics Division, Air Force Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson AFB OH, December 1978 (AFFDL-TR-78-170).
34. Venkayya, V. B. and V. A. Tischler. Discussion concerning large spacestructure model, April 1987.

35. Willner, Dieter. Observation and Control of Partially Unknown Systems. PhD dissertation. Electronic Systems Laboratory, Electrical Engineering Department, Massachusetts Institute of Technology, Cambridge MA, May 1973 (ESL-R-496).
36. Yip, Pak T. "Multiple Model Adaptive Filter for Tank Fire Control and Its Microprocessor Implementation," Transactions of the Conference of Army Mathematicians. Bethesda MD, 1982 (AD-P001 033).

VITA

Captain Robert W. Lashlee, Jr. was born April 4, 1960 in Camp Lejeune, North Carolina to Bob and Krista Lashlee. He graduated from high school in Beebe, Arkansas in 1978. He then went on to graduate with a BS in Chemistry from the University of Central Arkansas in December 1981. After receiving his commission from Officer's Training School in May 1982, he then attended the University of New Mexico at Albuquerque on an AFIT civilian institute program. In May 1984 he received a BS in Electrical Engineering, graduating with distinction. He was then assigned to the Guidance and Control Branch of the Air Force Armament Laboratory at Eglin Air Force Base in Florida, before coming to the Air Force Institute of Technology in May 1986. He has a wife, Tamee, and two children, Robert III and Tiffanee.

Permanent Address: 303 Pecan Street
Beebe, Arkansas 72012

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A192000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/87D-36			5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (if applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
		WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Moving-Bank Multiple Model Adaptive Estimation Applied To Flexible Spacestructure Control				
12. PERSONAL AUTHOR(S) Robert W. Lashlee, Jr., B.S.E.E., Captain, USAF				
13a. TYPE OF REPORT MS Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1987 December	15. PAGE COUNT 227	
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP		
			Adaptive Control Systems, Adaptive Filters, Multiple Model Adaptive Estimation, Multiple Model Adaptive Control	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
Title: Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control Thesis Chairman: Peter S. Maybeck, Professor of Electrical Engineering				
<div style="text-align: right;"> Approved for public release: 12W APR 19947. BY <i>W. E. Wilson</i> 14 JUN 88 EYNN E. WILSON DAIR for Public Release Air Force Institute of Technology (AFIT) Wright-Patterson AFB OH 45433 </div>				
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Peter S. Maybeck, Prof. of Elec. Eng.		22b. TELEPHONE (Include Area Code) (513) 255-2057		22c. OFFICE SYMBOL AFIT/ENG

UNCLASSIFIED

Abstract

This investigation focused on the use of moving-bank multiple model adaptive estimation and control (MMAE). Moving-bank MMAE reduces the computational burden of MMAE by implementing only a subset of the Kalman filters (9 filters versus 100 in this research) that are necessary to describe the system to be estimated/controlled. Important to the development of the moving-bank MMAE are the decision logics governing the selection of the subset of filters. The decision logics cover three situations: initial acquisition of unknown parameter values; tracking unknown parameter values; and reacquisition of the unknown parameters following a "jump" change in these parameter values.

This thesis applied moving-bank MMAE to a rotating two bay truss model of a flexible spacestructure. The rotating two bay truss approximated a space structure that had a hub with appendages extending from the structure. The mass of the hub was large relative to the mass of the appendage. The mathematical model was developed using finite element analysis, transformed into modal formulation, and reduced using a method referred to as singular perturbations. Multiple models were developed by assuming that variation occurred in the mass and stiffness of the structure.

Results showed that the primary cause for the increase in the performance of the moving-bank algorithm over past efforts was the choice of the measurement covariance noise matrix. This research effort also showed that the space discretization level can play an important role in the performance of the moving-bank MMAC; therefore, care must be taken in determining the discretization level. Finally, this research effort showed that the moving-bank MMAC provided increased control performance over a fixed bank controller with a discretization that covered the entire range of parameter variation. In addition, the moving-bank controller converged to the performance of a benchmark controller.

END
DATE
FILMED

4-88
DTIC